

---

## CAPITULO 5

### Estructura de la capa de datos



Arquitectura Tecnológica de Aplicaciones WEB

348

---

---

---

---

---

---

---

---

### Temario

- Modelo Entidad-Relación
- Estructura de un RDBMS
- Fronteras de las Bases de Datos
- Lenguaje SQL



Arquitectura Tecnológica de Aplicaciones WEB

349

---

---

---

---

---

---

---

---

## Modelo Entidad Relación



Arquitectura Tecnológica de Aplicaciones WEB

350

---

---

---

---

---

---

---

## Introducción

- **Modelo Entidad Relación.**
- **Desarrollado por E. Codd 1970.**
- **Modelo de datos basado en representación de registros.**
- **Hoy en día se trata del modelo más usado en aplicaciones comerciales.**
- **Se compone de:**
  - Estructura de datos.
  - Integridad de datos.
  - Manipulación de datos.



---

---

---

---

---

---

---

## Modelo Relacional

- El modelo relacional es una forma de ver los datos es decir, es una receta para representar los datos, mediante tablas, y la receta para manipular esa representación mediante operadores.
- El modelo relacional se preocupa de tres aspectos de los datos : su estructura, su integridad y su manipulación.
- Desde una visión histórica este modelo es relativamente nuevo, los primeros sistemas de bases de datos estaban basados en el modelo de redes o jerárquicos, orientados a una implementación física de la base de datos.
- Con la introducción del modelo relacional se desarrolló una teoría orientada a las bases de datos relacionales. Esta teoría ayuda al diseño de las bases de datos y al proceso de consultas del usuario.



---

---

---

---

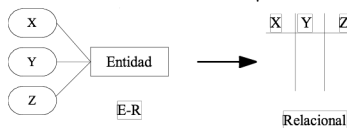
---

---

---

## Modelo Relacional (2)

- En este modelo la base de datos es vista por el usuario como una relación de tablas. Cada fila de una tabla es un registro o tupla y los atributos son columnas o campos.



- Cada una de las tablas de la base de datos debe tener un nombre único. Generalmente corresponde al nombre de la entidad.
- Cada columna tiene asociado un dominio que es el conjunto de valores posibles para esa columna.



---

---

---

---

---

---

---

## Modelo Relacional (3)

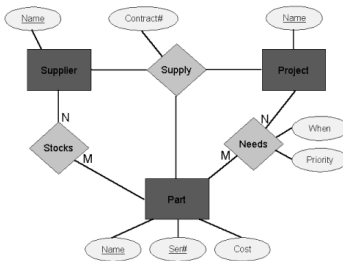
- El orden en que se listan las filas no tiene importancia.
- Si las columnas están rotuladas (tiene un nombre), entonces el orden de las columnas no tiene importancia.
- Representación de Conjuntos de Entidades

### Material

Código	Nombre	Valor	Cantidad
001	Cuaderno	500	1000
002	Papel Continuo	20	3000
003	Clip	15	1500

## Modelo relacional

- ¿Cómo se diseña la base de datos?
- Usando el modelo entidad relación, se crea un repositorio de datos que permite:
  - Contestar cualquier presunta sobre los datos.
  - Minimizar la redundancia



## El modelamiento de datos

- Características
  - Es un desarrollo Top-Down
  - La idea general es hacer una abstracción del negocio y llevarlo a una representación esquemática
- Una vez creado el modelo de datos, se usará una herramienta de software para implementarlo.
- Si el tipo de modelo es el Entidad Relación, se aconseja que la herramienta este orientada a ese tipo de estrategia.

## Modelo llevado a la base de datos

- Los SABDR (Sistemas Administradores de Bases de Datos Relacionales) proveen un lenguaje estandarizado para llevar el modelo relacional a una representación computacional
- Algunos SABDR poseen objetos que permiten definir reglas complejas del negocio, que han sido impuestas al modelo de datos.
- Una vez tomada la decisión respecto de cuál será el SABDR a usar, hay que considerar :
  - Desempeño
  - Reglas de integridad de datos
  - Integración con otros sistemas en desarrollo o en producción.
  - Documentación



Arquitectura Tecnológica de Aplicaciones WEB

357

---

---

---

---

---

---

---

---

## ¿Porqué usamos el modelo entidad relación?

- Es relativamente fácil de entender para la contraparte técnica o para un cliente, en comparación con las antiguas formas de modelamiento de datos.
- Elimina la redundancia de los datos.
- Cualquier consulta que por sobre los datos se realice, es posible de ser contestada.
- Ya está estandarizado y los desarrolladores lo entienden fácilmente.
- Permite dimensionar los requerimientos de hardware para la base de datos.



Arquitectura Tecnológica de Aplicaciones WEB

358

---

---

---

---

---

---

---

---

## Conceptos

- Entidad
  - Son los objetos que puedo caracterizar dentro del problema a modelar, por ejemplo clientes, vendedores, etc
- Atributo
  - Son las características que definan la entidad
  - Por ejemplo, en un cliente : Carné, edad, etc.
- Relación
  - Es la asociación directa que ocurre entre dos entidades.
  - Por ejemplo, en una escuela hay alumnos y profesores, que serían las entidades, la posible relación es " enseñá a "



Arquitectura Tecnológica de Aplicaciones WEB

359

---

---

---

---

---

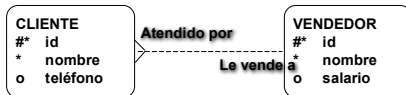
---

---

---

## Modelo Entidad Relación

- Ejemplo



- Situaciones

- "... Un vendedor le puede vender artículos a uno o más clientes ..."
- "... Algunos vendedores aun no han sido asignados a clientes ..."



## Convenciones

- Entidad

- Se encierra en una caja
- Posee un solo nombre, generalmente un sustantivo

- Atributo

- nombre en singular
- Si es indispensable, se coloca un "\*"
- Si es opcional, un "o"

- Relación

- Identificador único (UID)
- Llave primaria marcada con "#"
- Llave secundaria marcada con "(#)"



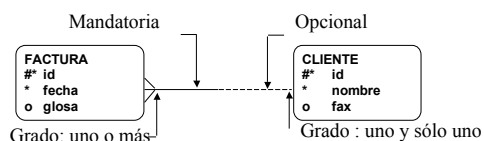
## Ejemplo

- Nomenclatura

- Las relaciones pueden ser del tipo mandatoria o del tipo opcional.

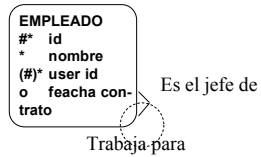
- Example

- Cada FACTURA debe ser de uno y sólo un CLIENTE.
- Cada CLIENTE puede tener una o más FACTURAS.



## Relación Recursiva

- Se trata de la relación que se genera entre la entidad y y si misma.
- Se representa a través de un bucle



Arquitectura Tecnológica de Aplicaciones WEB

363

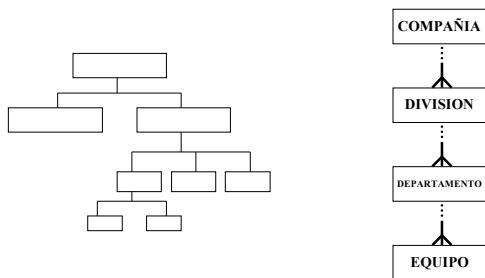
## Tipos de Relaciones

- Uno a uno
  - Define una relación biunívoca, por ej. Un computador tiene sólo una y sólo una Mother Board y una Mother Board está en un único computador
  - Ejemplo: el matrimonio (se supone)
- Muchos a uno
  - Muchas instancias de una entidad pueden ser atendidas por una sola instancia de otra entidad
  - Ejemplo: Las facturas y su detalle
- Muchos a mucho
  - Muchas instancias de una entidad son atendidas por muchas instancias de otras entidades.
  - Ejemplo: Cursos y alumnos.

Arquitectura Tecnológica de Aplicaciones WEB

364

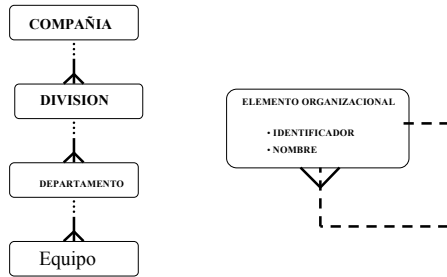
## Relación Jerárquica :



Arquitectura Tecnológica de Aplicaciones WEB

365

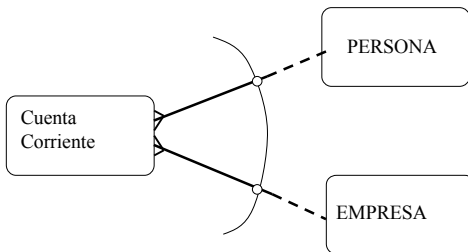
## Otra representación de la jerarquía



Arquitectura Tecnológica de Aplicaciones WEB

366

## Relación : ARCOS



Arquitectura Tecnológica de Aplicaciones WEB

367

## ATRIBUTOS

Obtenga todos los Atributos Necesarios y Suficiente

CLIENTE

- # \* RUT
- \* NOMBRE
- o DIRECCION
- \* FONO
- o FAX
- o RANKING

# : UID

\* : OBLIGATORIO

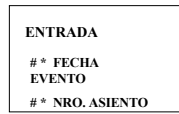
o : OPCIONAL

Arquitectura Tecnológica de Aplicaciones WEB

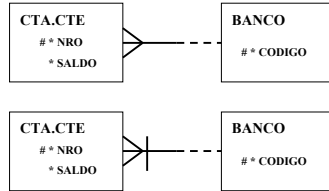
368

## Atributos : UID

COMPUESTA



A TRAVES DE LA RELACION



Arquitectura Tecnológica de Aplicaciones WEB

369

---

---

---

---

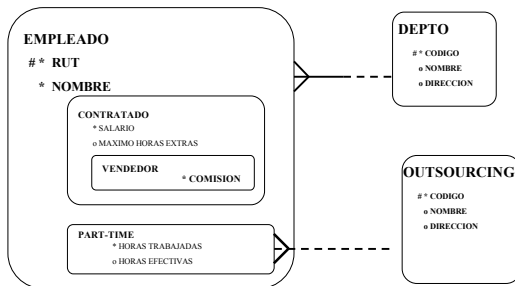
---

---

---

---

## Atributos : Superclases y relaciones



Arquitectura Tecnológica de Aplicaciones WEB

370

---

---

---

---

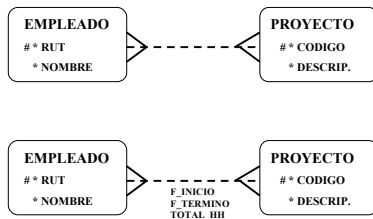
---

---

---

---

## Relación con atributos



Arquitectura Tecnológica de Aplicaciones WEB

371

---

---

---

---

---

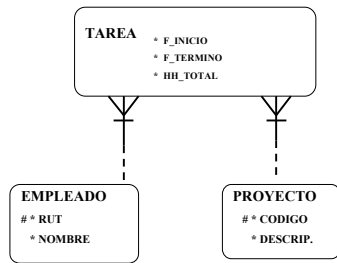
---

---

---



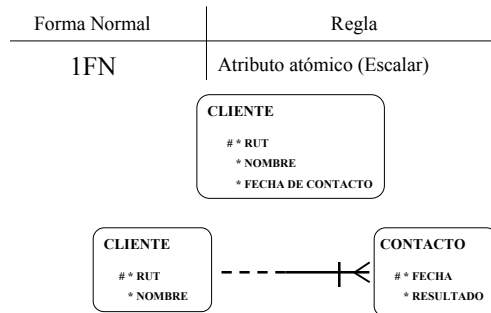
## Relación con atributos



Arquitectura Tecnológica de Aplicaciones WEB

372

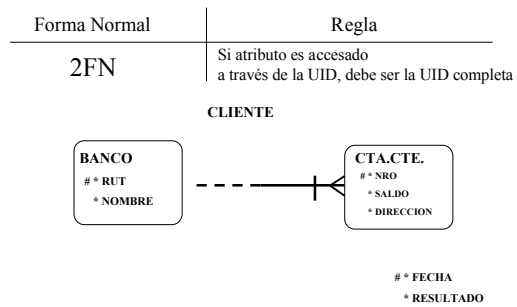
## NORMALIZACION



Arquitectura Tecnológica de Aplicaciones WEB

373

## NORMALIZACION

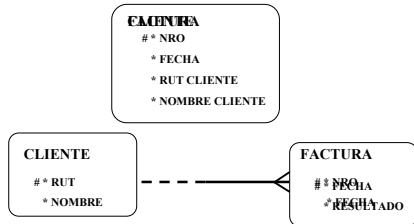


Arquitectura Tecnológica de Aplicaciones WEB

374

## NORMALIZACION

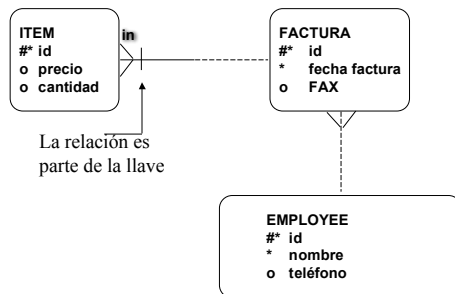
Forma Normal	Regla
3FN ... B-C	Un atributo que no es parte de la UID solo puede ser accesado a través de la UID



Arquitectura Tecnológica de Aplicaciones WEB

375

## Ejemplo del MER



Arquitectura Tecnológica de Aplicaciones WEB

376

## Restrictores de integridad (constraint)

- Aseguran la consistencia de los datos
- Pueden ser hechos a nivel de la Base de Datos o en la aplicación
- Son llaves primarias, foráneas, etc.

Tipo	
Entidad	Aquellos campos que no forman parte de la PK pueden ser NULL
Referencial	No hay detalles sin maestros
Columna	Los datos deben ser del mismo tipo que la columna
otros	Triggers de bases de datos

Arquitectura Tecnológica de Aplicaciones WEB

377

## Llave primaria ( Primary Key :PK)

- Aseguran la NO existencia de filas duplicadas-
- Se trata de un atributo o conjunto de ellos, que definen en forma única a una fila.
- Implícitamente, definen una estructura de datos que permita asegurar la NO repetición de datos.
- Ejemplo: En la entidad ALUMNO, el número de matrícula identifica en forma única a un alumno.



---

---

---

---

---

---

---

---

## Llave foránea (Foreign Key :FK)

- Es una columna o conjunto de estas, que referencian a una PK o UK en la misma tabla o en otra.
- Sirven para definir relaciones tales como las master-detail
- El valor contenido en la FK debe calcar con el valor de la PK que referencia o ser NULL
- Si una FK es parte de una PK, entonces no puede contener NULL.
- Ejemplo : No se puede borrar un maestro, sin haber borrado primero el detalle.



---

---

---

---

---

---

---

---

## FK Ejemplo

La columna dept\_id es la FK en la tabla EMPLEADO y referencia a la columna id en la tabla DEPT.

Tabla EMPLEADO

ID	Apellido	Fono	...	DEPT_ID	...
1	Velasquez	9999		50	
2	Ngao	555		41	
3	Nagayama	888		31	
4	Quick-To-See	9998		10	
5	Ropeburn	222		50	

PK

FK

Tabla DEPT

ID	NAME	Loc
10	Finanzas	1
31	Ventas	1
41	Comercial	1
50	Gestion	1

PK



---

---

---

---

---

---

---

---

## El MER llevado a la B.D.

- Las entidades definidas, se transforman en tablas, a partir de comandos que interpreta el SABDR.
- El nombre que tienen las tablas, coincide con el de las relaciones.
- Hay que crear objetos adicionales de apoyo, tales como :
  - Indices
  - Triggers
  - Vistas
  - Restrictores



Arquitectura Tecnológica de Aplicaciones WEB

381

---

---

---

---

---

---

---

---

## Ejercicios

Una facultad posee alumnos, los cuales toman varios cursos durante un semestre. Los profesores que dictan los ramos, pertenecen a uno y sólo un departamento dentro de la facultad y pueden dictar más de un curso durante el semestre.

Es información importante de los alumnos, su nombre, matrícula, año ingreso, fecha de nacimiento, sexo y un registro de todos los cursos que ha tomado, registrando si los aprobó o no y en que semestre.

Los cursos que toma un alumno, tienen un código, un nombre y son dictado por un departamento en específico. Poseen además, un identificador de sección, por cuanto puede ser dictado más de una vez en el semestre, pero solo por un profesor a la vez.

Para tomar un curso, el alumno debe haber cursado previamente otros y haberlos aprobado.

Establezca un modelo entidad relación para el problema anteriormente planteado. En caso de ser necesario, haga los supuestos que estime convenientes.



Arquitectura Tecnológica de Aplicaciones WEB

382

---

---

---

---

---

---

---

---

## Estructura de un RDBMS



Arquitectura Tecnológica de Aplicaciones WEB

383

---

---

---

---

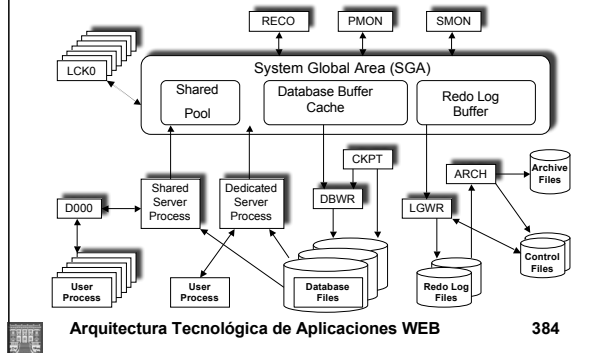
---

---

---

---

## Estructura: Caso Oracle



## Estructuras de Memoria

### System Global Area (SGA)



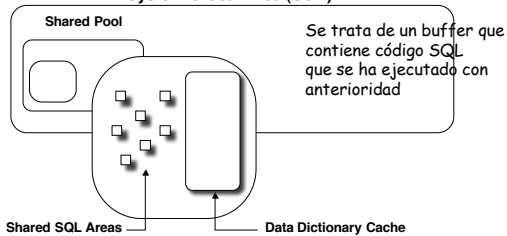
Se trata de un trozo continuo de la memoria RAM del computador, donde se definen buffers de memoria que usará Oracle para almacenar datos, código y otras estructuras de datos

Arquitectura Tecnológica de Aplicaciones WEB

385

## El Shared Pool

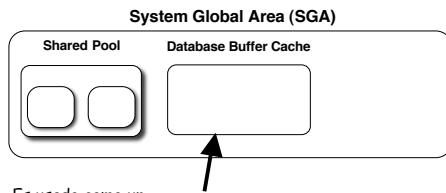
### System Global Area (SGA)



Arquitectura Tecnológica de Aplicaciones WEB

386

## El Database Buffer



Es usado como un buffer de datos. Cada vez que se ejecuta una instrucción que involucre la obtención de datos, primero se busca en el cache y si no están, en los discos.



---

---

---

---

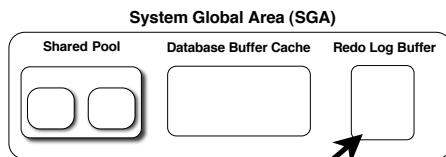
---

---

---

---

## El Redo Log Buffer



Corresponde a una bitácora, es decir, un registro de los comandos que se han ejecutado. Sirve para recuperar la base de datos ante una posible falla



---

---

---

---

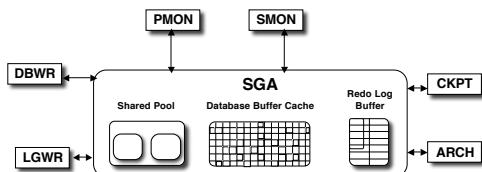
---

---

---

---

## Procesos Background



Se trata de procesos "demonios" que siempre están corriendo y parten cuando se "levanta" el Oracle. Cuatro son los esenciales, es decir, si no están, hay problemas. Estos son : PMON, SMON LGWR y DBWR



---

---

---

---

---

---

---

---

## Process Monitor (PMON)

- Se encarga de eliminar los recursos tomados por conexiones que han terminado anormalmente
- Deshace las transacciones que terminaron anormalmente.
- Libera todo tipo de locks, cuando el proceso cliente se desconecta de la instancia.



Arquitectura Tecnológica de Aplicaciones WEB

390

---

---

---

---

---

---

---

## System Monitor (SMON)

- Apoya a la recuperación de la instancia.
- Asigna los espacios necesarios para realizar actividades de **join** y **sort**.
- Desfragmenta un archivo de datos.



Arquitectura Tecnológica de Aplicaciones WEB

391

---

---

---

---

---

---

---

## Data Base Writer (DBWR) y Log Writer (LGWR)

- El DBWR se encarga del traspaso de los datos desde RAM a disco y vice versa.
- El LGWR se encarga de llevar los la bitácora almacenada en el buffer de redo log a disco



Arquitectura Tecnológica de Aplicaciones WEB

392

---

---

---

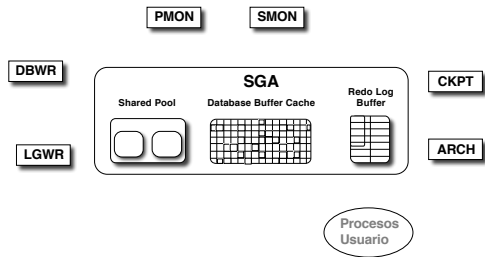
---

---

---

---

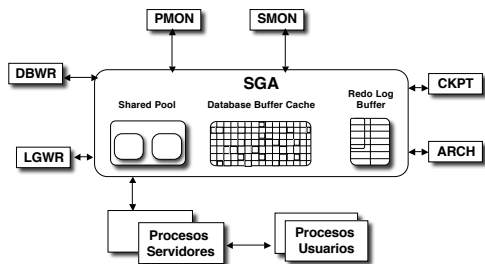
## Procesos usuario



Arquitectura Tecnológica de Aplicaciones WEB

393

## Procesos servidores



Arquitectura Tecnológica de Aplicaciones WEB

394

## Fronteras de las BD

Arquitectura Tecnológica de Aplicaciones WEB

395



## Definición

- **Database:** "Un conjunto logicamente coherente de datos relacionados, construido para una cierta aplicacion"
- **Database management system (DBMS)** es un software que permite a las bases de datos ser definidas, construidas y operarlas.
- **VLDB.** Very Large Data Base. Bases de datos muy grandes, contienen varios Tera Bytes (1000 Gigas).



---

---

---

---

---

---

---

---

## Para que usar los RDBMS?

- Hacer un cambio en el programa es mas caro que hacer un cambio en los datos.
- Los RDBMS son mas flexibles.
- Las consultas son mas fáciles.
- Son los rdbms mas rápidos que los stream system??



Copyright © 1995 United Feature Syndicate, Inc.  
Reproduction in whole or in part prohibited.



---

---

---

---

---

---

---

---

## Consultas (Queries)

- Todas las tuplas de la tabla Bike:  
`Select Color, Serial Number, Number of Gears  
From Bike;`
- Aplicando una condición para ser mas selectivo:

```
Select Color, Serial Number, Number of Gears  
From Bike  
Where Color = "Blue" and Number of Gears >= 10;
```



---

---

---

---

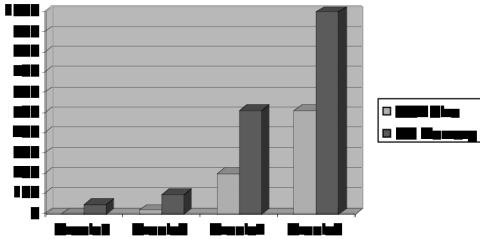
---

---

---

---

## El SGA: Oracle



Arquitectura Tecnológica de Aplicaciones WEB

399

## VLDB: El administrador de bases de datos en problemas!!

- El administrador de BD se instala sobre el sistema operativo.
- Problemas con el almacenamiento. Que pasa si el SO se cae??
- Nuevos tipos de datos.
- Nuevos lenguajes de mantencion de datos.
- Nueva generacion de hardware y software.

Arquitectura Tecnológica de Aplicaciones WEB

400

## Problemas de Almacenamiento : Redundant Array of Inexpensive Disks (RAID)

- Es una forma de almacenar datos distribuidos en varios discos fisicos. Se usan tecnicas como Disk striping (RAID nivel 0) y espejado de disco (RAID nivel 1).
- Hay 6 configuraciones de RAID posibles.
- Se tienen muchas aplicaciones posibles, particularmente en el ambiente de negocios, donde los datos no pueden perderse.

Arquitectura Tecnológica de Aplicaciones WEB

401

## RAID: Redundant Array of Inexpensive Disks (2)

- Muchas empresas no pueden que sus sistemas se encuentren caídos ante el evento de una falla de disco. Ellos entonces requieren grandes subsistemas de almacenamiento con capacidades del orden de los TeraBytes.
- Otras trabajan con archivos multimediales que requieren una alta tasa de transferencia, la cual excede lo que los discos normales pueden dar.
- Los principios fundamentales detras de RAID es el uso de multiples discos duros ordenados en arreglos, los cuales se comportan como un solo gran disco.

---

---

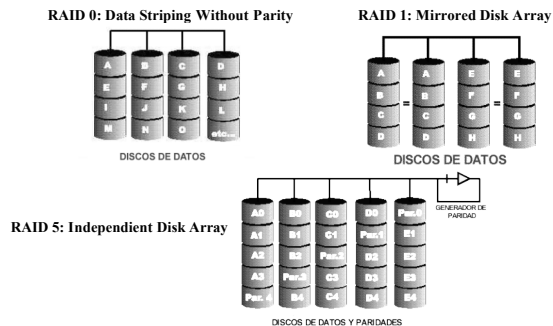
---

---

---

---

## RAID: Ejemplos



---

---

---

---

---

---

## Datos, Información y Conocimiento

- Que pasa cuando una institucion tiene una gran cantidad de datos historicos?
- Son importantes los datos historicos para continuar el negocio?
- Tecnologias de almacenamiento: Son posibles las bases de datos muy grandes?
- Entonces, los datos son importantes, pero el proceso del negocio requiere informacion y conocimiento!!

---

---

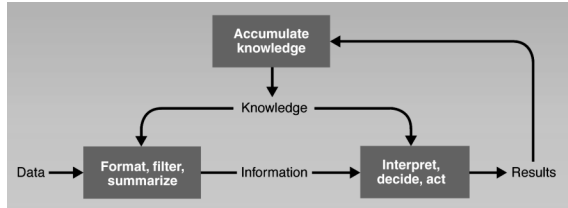
---

---

---

---

## Relaciones entre: Datos, Información y Conocimiento



Arquitectura Tecnológica de Aplicaciones WEB

405

## Mirada historica al procesamiento de la informacion

El objetivo del procesamiento de la informacion es transformar los datos en informacion!

### Porque?

Porque las preguntas en los negocios se hacen el en ambito de la *informacion* y el *conocimiento* de como aplicar esta informacion en la resolucion de un problema.

Arquitectura Tecnológica de Aplicaciones WEB

406

## El proceso de KDD: Motivaciones

- "The know how is in house"
- En un supermercado: cuales son los productos correlacionados?
- Pañales y Cerveza?
- Problema de la inflacion de datos:
  - Herramientas de recoleccion automatica de datos, disponibilidad de dispositivos de almacenamiento a precios mas baratos lleva a una tremenda cantidad de datos almacenados, data warehouses y otros repositorios de informacion.
- Estamos llenos de datos, pero sedientos de conocimiento!!!
  - Datos hay en todas partes
  - Entender y usar esos datos es una tarea inminente!
- Solucion: *Knowledge Discovery in Data Bases*

Arquitectura Tecnológica de Aplicaciones WEB

407

## Knowledge Discover in Data Bases

- "Es el proceso de extraccion no-trivial de informacion de los datos, informacion que esta implicitamente presente en los datos, anteriormente desconocida y potencialmente util para el usuario"
- Very Large Databases (VLDB) han llegado a ser un estandar de la industria. Permitiendo asi minar los datos manualmente.
- Las herramientas automatizadas han sido requeridas para ayudar a la extraccion de esos patrones.



---

---

---

---

---

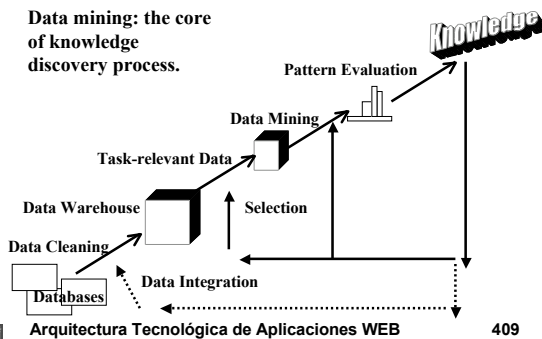
---

---

---

## El proceso de KDD

Data mining: the core of knowledge discovery process.



---

---

---

---

---

---

---

---

## Data Warehouse and Data Mart

- "Data Warehouse is a subject-oriented, integrated, nonvolatile, time-variant collection of data in support of management's decisions". Bill Inmon.
- Data mart is a collection of subject areas organized for decision support based on the needs of a given department.



---

---

---

---

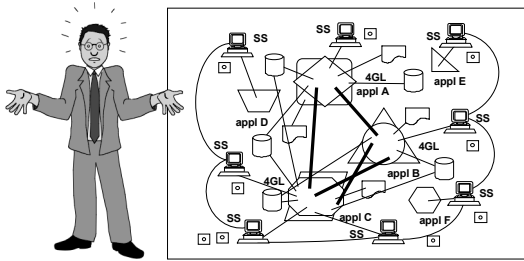
---

---

---

---

## I have data but where do I go?"

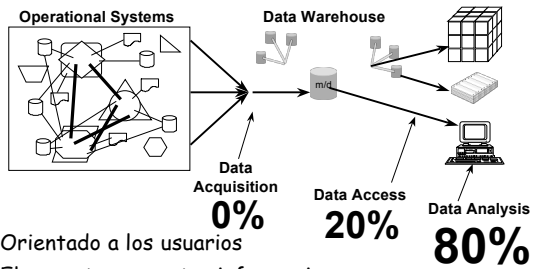


Tenemos datos, pero es difícil encontrar información útil.

Arquitectura Tecnológica de Aplicaciones WEB

411

## DW Goal!

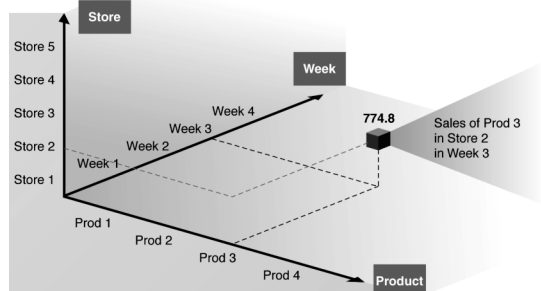


- Orientado a los usuarios
- El experto encuentra información y genera conocimiento.

Arquitectura Tecnológica de Aplicaciones WEB

412

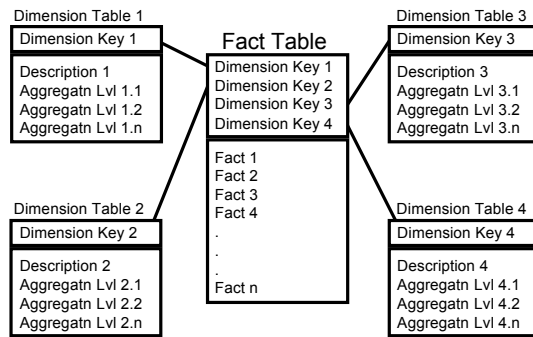
## Building DW: Multidimensional Analysis



Arquitectura Tecnológica de Aplicaciones WEB

413

## DW: The Star Schema



Arquitectura Tecnológica de Aplicaciones WEB

414

## Dimension Table

Dimension Table 1
Dimension Key 1
Description 1
Aggregatn Lvl 1.1
Aggregatn Lvl 1.2
Aggregatn Lvl 1.n

- Describe los datos
- Las llaves deben tener el maximo de detalle posible (e.g. State vs. ZIP Code) o de lo contrario usar :
- Las llaves no primarias pueden ser usadas, pero bajan la utilidad de las tablas
- Se manejan los niveles de agregacion

Arquitectura Tecnológica de Aplicaciones WEB

415

## Fact Table

### Fact Table

Dimension Key 1
Dimension Key 2
Dimension Key 3
Dimension Key 4
Fact 1
Fact 2
Fact 3
Fact 4
.
.
Fact n

- Cuantifican los datos que se encuentran en las tablas de Dimensiones
- Key made up of unique combination of values of dimension keys
  - ALWAYS contains date or date dimension
- Fact values should be additive
  - Aggregations of quantities or amounts from atomic level
  - No percentages or ratios
  - May be non-additive, time-variant data

Arquitectura Tecnológica de Aplicaciones WEB

416

## New Technology: Partitioning Tables

- Tables can now be split into thousands of pieces.
  - Using partition tables and indexes
  - Only a subset of the data is queried
  - All of the data *COULD* be queried
  - Leads to enhanced performance of large tables
  - Partitioned views was the precursor to this
  - Data Warehouses can be tuned greatly!
  - Re-orgs can be done on a partition level



Arquitectura Tecnológica de Aplicaciones WEB

417

---

---

---

---

---

---

---

---

## Partitioning Tables: Example

```
CREATE TABLE DEPT
(DEPTNO          NUMBER(2),
 DEPT_NAME       VARCHAR2(30))
PARTITION BY RANGE(DEPTNO)
(PARTITION D1 VALUES LESS THAN (10) TABLESPACE DEPT1,
 PARTITION D2 VALUES LESS THAN (20) TABLESPACE DEPT2,
 PARTITION D3 VALUES LESS THAN (MAXVALUE) TABLESPACE DEPT3);

INSERT INTO DEPT VALUES (1, 'DEPT 1');
INSERT INTO DEPT VALUES (7, 'DEPT 7');
INSERT INTO DEPT VALUES (10, 'DEPT 10');
INSERT INTO DEPT VALUES (15, 'DEPT 15');
INSERT INTO DEPT VALUES (22, 'DEPT 22');
```



Arquitectura Tecnológica de Aplicaciones WEB

418

---

---

---

---

---

---

---

---

## The Parallel Query Option

- Used on CPU intensive jobs.
- Allows the query to be spread across multiple CPU's
- Short jobs will usually suffer from this option because of the time required to divide and reassemble the query.
- Available only when a Full table scan or a Sort operation is being performed EXCEPT on Partitioned Tables and Indexes.



Arquitectura Tecnológica de Aplicaciones WEB

419

---

---

---

---

---

---

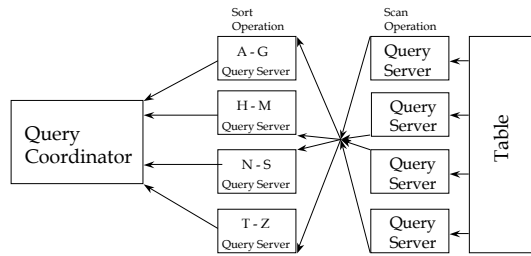
---

---



## Degree of Parallelism

A query with a degree of 4 could also look like this



Arquitectura Tecnológica de Aplicaciones WEB

420

## Comercial RDBMS and Data Warehouse

- IBM DB2.
- Informix Dynamic server 7.3
- Sybase
- Oracle 10G
- Microsoft SQL Server.

Arquitectura Tecnológica de Aplicaciones WEB

421

## New Problem: New Data Types

- Sound, movies, pictures, text, etc.
- Simple search is a no trivial problem.
- The storage problem is under control.
- The maintenance of the data is a real problem.
- The Entity Relational Model is changing. The new model considerate Object Oriented Design.
- The new model need special hardware structure.

Arquitectura Tecnológica de Aplicaciones WEB

422

## Data Mining

- "Data analysis in order to discover hidden correlations (pattern rules) in huge data sets"
- "Data mining is the process of extracting valid, previously unknown, comprehensible and actionable information from large data bases"



Arquitectura Tecnológica de Aplicaciones WEB

423

---

---

---

---

---

---

---

---

## DM: algorithm and components

- Model representation
    - Descriptions of discovered patterns
    - Overly limited representation
    - Model evaluation criteria
    - How well a pattern (model) meets goals (fit function)
    - eg., accuracy, novelty, etc.
  - Search method
    - parameter search: optimization of parameters for a given model representation
    - model search: considers a family of models
- Different methods suit different problems. Proper problem formulation crucial.



Arquitectura Tecnológica de Aplicaciones WEB

424

---

---

---

---

---

---

---

---

## Data Mining: Models

- Prediction Methods
  - Using some variables to predict unknown or future values of other variables
- Descriptive Methods
  - Finding human-interpretable patterns describing the data.



Arquitectura Tecnológica de Aplicaciones WEB

425

---

---

---

---

---

---

---

---

## Data Mining Tasks

- Classification
- Clustering
- Association Rule Discovery
- Sequential Pattern Discovery
- Regression
- Deviation Detection



---

---

---

---

---

---

---

---

## Association Rules

- Should be used for prediction with great caution; rules do not really reflect causality.
- If users buy pencils whenever they buy paper, we have the rule {pencil} => {paper}, but there is no causal link; offering a promotion on pencils in order to stimulate sales of paper will be a failure!



---

---

---

---

---

---

---

---

## Decision Tree

- An internal node represents a test on an attribute.
- A branch represents an outcome of the test, e.g., Color=red.
- A leaf node represents a class label or class label distribution.
- At each node, one attribute is chosen to split training examples into distinct classes as much as possible
- A new case is classified by following a matching path to a leaf node.



---

---

---

---

---

---

---

---

## Clustering

- Clustering: Given points in some space, group the points into a small number of clusters, each cluster consisting of points that are "near" in some sense.
- The key: Distance measures!!



Arquitectura Tecnológica de Aplicaciones WEB

429

---

---

---

---

---

---

---

---

## Clustering and Classification

- Supervised learning (classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations
  - New data is classified based on the training set
- Unsupervised learning (clustering)
  - The class labels of training data are unknown
  - Given a set of measurements, observations, etc. with the aim of establishing the existence of classes or clusters in the data



Arquitectura Tecnológica de Aplicaciones WEB

430

---

---

---

---

---

---

---

---

## Data Mining Research Trend

- Text mining, text database and information retrieval.
- Multimedia data mining.
- OLAM (OLAP Mining).
- Web mining (Data Mining and WWW)
  - E-commerce
  - Information retrieval (search)
  - Network management



Arquitectura Tecnológica de Aplicaciones WEB

431

---

---

---

---

---

---

---

---

## Mine the Web!!

- Web: A huge, widely-distributed, highly heterogeneous, semi-structured, hypertext/hypermedia, interconnected, evolving information repository.
- Web is a huge collection of documents plus
  - Hyper-link information
  - Access and usage information
- Enormous wealth of information on Web
  - Financial information (e.g. stock quotes)
  - Book/CD/Video stores (e.g. Amazon)
  - Restaurant information (e.g. Zagats)
  - Car prices (e.g. Carpoint)
- Lots of data on user access patterns
  - Web logs contain sequence of URLs accessed by users



---

---

---

---

---

---

---

---

## Why is Web Mining Different?

- The Web is a huge collection of documents except for
  - Hyper-link information
  - Access and usage information
- The Web is very dynamic
  - New pages are constantly being generated
- Complexity of Web pages: far greater than text document collection
- Develop new Web mining algorithms and adapt traditional data mining algorithms to
  - Exploit hyper-links and access patterns
  - Be incremental



---

---

---

---

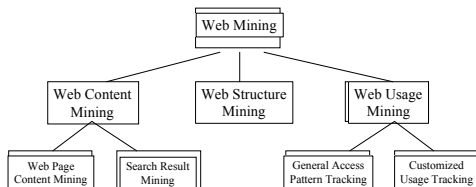
---

---

---

---

## Types of Web Mining



---

---

---

---

---

---

---

---

## Web Mining.

- Web Content Mining. Identify information within web pages.
- Web Structure Mining. Analysis of the web pages and hyperlinks between pages.
- Web Usage Mining. Understand access patterns and trends using web log data.



Arquitectura Tecnológica de Aplicaciones WEB

435

---

---

---

---

---

---

---

## A case: VLDB and KDD in Astronomy

- Astronomy has become an immensely data-rich field (and growing).
- There is a need for powerful DM/KDD tools
- There are excellent opportunities for interdisciplinary collaborations/partnerships



Arquitectura Tecnológica de Aplicaciones WEB

436

---

---

---

---

---

---

---

## VLDB: Virtual Observatory (VO) Concept

- A response of the astronomical community to the scientific and technological challenges posed by massive data sets
- Federate the existing and forthcoming large digital sky surveys and archives, and provide the tools for their scientific exploitation
- A dynamical, interactive, web-based research environment for the new astronomy with massive data sets
- Technology-enabled, but science-driven



Arquitectura Tecnológica de Aplicaciones WEB

437

---

---

---

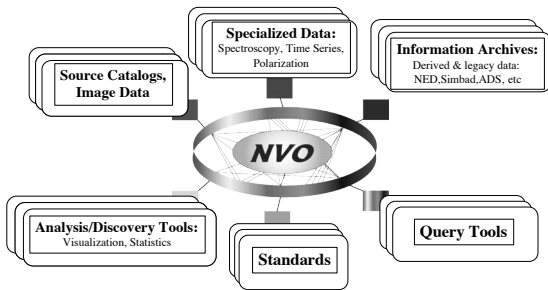
---

---

---

---

## NVO: The Concept



Source: <http://www.astro.caltech.edu/~george/vol/>

Arquitectura Tecnológica de Aplicaciones WEB

438

## The Technological Problems

- **Data Handling:**
  - Efficient database architectures/query mechanisms
  - Archive interoperability, standards, metadata ...
  - Survey federation (in the image and catalog domains) ... etc.
- **Data Analysis:**
  - **Data mining / KDD tools and services** (clustering analysis, anomaly and outlier searches, multivariate statistics...)
  - Visualization (image and catalog domains, high dimensionality parameter spaces) ... etc.

NB: A typical (single survey) catalog may contain  $\sim 10^9$  data vectors in  $\sim 10^2$  dimensions  
Terascale computing

Arquitectura Tecnológica de Aplicaciones WEB

439

## Introducción a SQL

Arquitectura Tecnológica de Aplicaciones WEB

440

## Objetivos

- Propósito e importancia de SQL.
- Como recuperar datos desde una Base de Datos usando SELECT :
  - Uso de la componente WHERE para agregar condiciones a las Consultas.
  - Ordenar los resultados de las Consultas usando ORDER BY.
  - Uso de Funciones Adicionales.
  - Agrupar datos usando GROUP BY y HAVING.
  - Uso de Sub-Consultas.



Arquitectura Tecnológica de Aplicaciones WEB

441

---

---

---

---

---

---

---

---

## Objetivos

- Usar Join entre tablas.
- Operaciones de Conjuntos :
  - UNION, INTERSECT, EXCEPT
- Como realizar transacciones sobre la Base de Datos usando :
  - INSERT, UPDATE, and DELETE.
- Tipos de datos soportados por SQL-92.
- Como crear y borrar tablas.



Arquitectura Tecnológica de Aplicaciones WEB

442

---

---

---

---

---

---

---

---

## Objetivos de SQL

- Un lenguaje de base de datos debe permitir :
  - Crear las estructuras de la Base de datos y sus relaciones.
  - Realizar inserción, modificación y borrado de datos.
  - Realizar consultas simples y complejas.
- Estas tareas se deben realizar con un mínimo esfuerzo del usuario.
- La estructura y sintaxis de comandos debe ser fácil de aprender.
- Debe ser portable.



Arquitectura Tecnológica de Aplicaciones WEB

443

---

---

---

---

---

---

---

---



## Objetivos

- Se conforma con palabras inglesas estándares.:

```
CREATE TABLE staff (sno VARCHAR(5),
                    lname VARCHAR(15),
                    salary DECIMAL(7,2));
INSERT INTO staff
VALUES ('SG16', 'Brown', 8300);
SELECT sno, lname, salary
FROM staff
WHERE salary > 10000;
```



---

---

---

---

---

---

---

---

## Historia de SQL

- En 1974, en el Laboratorio de IBM en San Jose D. Chamberlin definió un lenguaje llamado 'Structured English Query Language' o SEQUEL.
- Una versión mejorada de SEQUEL/ fue definida en 1976, pero su nombre fue cambiado por razones legales SQL.
- IBM produjo posteriormente un DBMS prototipo llamado System R, basado en SEQUEL/2
- En 1987, ANSI e ISO publicaron un estándar inicial para SQL.
- En 1989, ISO publicó una adición que definió ' una característica de realce de la integridad '.
- En 1992, la primera revisión importante al estándar de ISO ocurrió, designado SQL2 o SQL/92.



---

---

---

---

---

---

---

---

## Sentencia SELECT

```
SELECT [DISTINCT | ALL]
{* | [column_expression [AS new_name]] [...]}
FROM table_name [alias] [, ...]
[WHERE condition]
[GROUP BY column_list] [ H A V I N G
condition]
[ORDER BY column_list]
```



---

---

---

---

---

---

---

---

## Sentencia SELECT

**FROM** Especifica las tablas que se usaran  
**WHERE** Establece los filtros.  
**GROUP BY** Permite agrupar los datos.  
**HAVING** Permite generar filtros sobre los grupos de datos.  
**SELECT** Especifica las columnas que se consultaran.  
**ORDER BY** Especifica el orden de los datos.



## Ejemplo1 Todas las Columnas y Filas

Listar todos los datos de la tabla Staff.

```
SELECT sno, fname, lname, address, tel_no,  
       position, sex, dob, salary, nin, bno  
FROM staff;
```

- Se puede usar \* como abreviación para todas las columnas :

```
SELECT * FROM staff;
```



## Ejemplo1 Todas las Columnas y Filas

Table 13.1 Result table for Example 13.1.

sno	fname	lname	address	tel_no	position	sex	dob	salary	nin	bno
SL21	John	White	19 Taylor St, Cranford, London	0171-884-5112	Manager	M	1-Oct-45	30000.00	WK442011B	B5
SG37	Ann	Beech	81 George St, Glasgow PA1 2JR	0141-848-3345	Sen Asst	F	10-Nov-60	12000.00	WL422514C	B3
SG14	David	Ford	63 Ashby St, Patrick, Glasgow G11	0141-339-2177	Deputy	M	24-Mar-58	18000.00	WL220658D	B3
SA9	Mary	Howe	2 Elm Pl, Aberdeen AB2 3SU		Assistant	F	19-Feb-70	9000.00	WMS52187D	B7
SG5	Susan	Reed	5 Cl Western Rd, Glasgow G12	0141-334-2001	Manager	F	3-Jun-40	24000.00	WKS88932E	B3
SL41	Jillie	Lee	28 Malvern St, London NW2	0181-554-3541	Assistant	F	13-Jun-65	9000.00	WA280575K	B5

(6 rows)



## Ejemplo2 Uso de DISTINCT

Lista los numeros de propiedad de todas las propiedades vistas.

```
SELECT pno
FROM viewing;
```

Table 13.3(a) Result table for Example 13.3 with duplicates.

<i>pno</i>
PA14
PG4
PG4
PA14
PG36

(5 rows)



---

---

---

---

---

---

---

---

## Ejemplo2 Uso de DISTINCT

- Use DISTINCT para eliminar los codigos duplicados:

```
SELECT DISTINCT pno
FROM viewing;
```

Table 13.3(b) Result table for Example 13.3 with duplicates eliminated.

<i>pno</i>
PA14
PG4
PG36

(3 rows)



---

---

---

---

---

---

---

---

## Ejemplo3 Campos Calculados

Produce una lista con los sueldos mensuales de todos los empleados.

```
SELECT sno, fname, lname, salary/12
FROM staff;
```



---

---

---

---

---

---

---

---

### Ejemplo3 Campos Calculados

Table 13.4 Result table for Example 13.4.

sno	fname	lname	col4
SL21	John	White	2500.00
SG37	Ann	Beech	1000.00
SG14	David	Ford	1500.00
SA9	Mary	Howe	750.00
SG5	Susan	Brand	2000.00
SL41	Julie	Lee	750.00

(6 rows)

• Par n AS:

```
SELECT sno, fname, lname,  
       salary/12 AS monthly_salary  
FROM staff;
```



### Ejemplo4 Condición de Búsqueda

Listar todos los empleados con un sueldo mayor a 10.000.

```
SELECT sno, fname, lname, position, salary  
FROM staff  
WHERE salary > 10000;
```



### Ejemplo4 Condición de Busqueda

Table 13.5 Result table for Example 13.5.

sno	fname	lname	position	salary
SL21	John	White	Manager	30000.00
SG37	Ann	Beech	Snr Asst	12000.00
SG14	David	Ford	Deputy	18000.00
SG5	Susan	Brand	Manager	24000.00

(4 rows)



## Ejemplo5 Miembro de un Conjunto

Listar a todos los encargados y directores adjuntos.

```
SELECT sno, fname, lname, position
FROM staff
WHERE position IN ('Manager', 'Deputy');
```



---

---

---

---

---

---

---

## Ejemplo5 Miembro de un Conjunto

Table 13.8 Result table for Example 13.8.

<i>sno</i>	<i>fname</i>	<i>lname</i>	<i>position</i>
SL21	John	White	Manager
SG14	David	Ford	Deputy
SG5	Susan	Brand	Manager

(3 rows)



---

---

---

---

---

---

---

## Ejemplo5 Miembro de un Conjunto

- Existe una versión con negación (NOT IN).
- IN no agrega funcionalidades nuevas ya que se podría conseguir lo mismo con :

```
SELECT sno, fname, lname, position
FROM staff
WHERE position='Manager' OR position='Deputy';
```

- IN es más eficiente con mas elementos.



---

---

---

---

---

---

---

## Ejemplo6 Comparando Patrones

Encontrar todos los empleados con la palabra 'Glasgow' en su dirección.

```
SELECT sno, fname, lname, address, salary
FROM staff
WHERE address LIKE '%Glasgow%';
```



Arquitectura Tecnológica de Aplicaciones WEB

459

---

---

---

---

---

---

---

---

## Ejemplo6 Comparando Patrones

Table 13.9 Result table for Example 13.9.

sno	fname	lname	address	salary
SG37	Ann	Beech	81 George St, Glasgow PA1 2JR	12000.00
SG14	David	Ford	63 Ashby St, Partick, Glasgow G11	18000.00
SG5	Susan	Brand	5 Gt Western Rd, Glasgow G12	24000.00

(3 rows)



Arquitectura Tecnológica de Aplicaciones WEB

460

---

---

---

---

---

---

---

---

## Ejemplo7 Comparando Patrones

- SQL tiene dos patrones especiales para la comparación de símbolos:
  - %: secuencia de 0 o mas caracteres;
  - \_ (underscore): cualquier caracter simple.
- LIKE '%Glasgow%' significa una secuencia de caracteres de algún largo que contiene 'Glasgow'.



Arquitectura Tecnológica de Aplicaciones WEB

461

---

---

---

---

---

---

---

---

## Ejemplo7 Ordenamiento por múltiples columnas

**Table 13.12(a)** Result table for Example 13.12 with one sort key.

<i>pno</i>	<i>type</i>	<i>rooms</i>	<i>rent</i>
PL94	Flat	4	400
PG4	Flat	3	350
PG36	Flat	3	375
PG16	Flat	4	450
PA14	House	6	650
PG21	House	5	600

(6 rows)



---

---

---

---

---

---

---

---

## Ejemplo8 Ordenamiento por múltiples columnas

- Cuatro columnas en la lista - sin un orden descendiente especificado -el sistema ordenará las filas en el orden en que él escoja.
- Ordenar por orden de renta, específicamente en orden descendiente:

```
SELECT pno, type, rooms, rent
FROM property_for_rent
ORDER BY type, rent DESC;
```



---

---

---

---

---

---

---

---

## Ejemplo8 Multiple Column Ordering

**Table 13.12(b)** Result table for Example 13.12 with two sort keys.

<i>pno</i>	<i>type</i>	<i>rooms</i>	<i>rent</i>
PG16	Flat	4	450
PL94	Flat	4	400
PG36	Flat	3	375
PG4	Flat	3	350
PA14	House	6	650
PG21	House	5	600

(6 rows)



---

---

---

---

---

---

---

---

## SELECT Sentencias agregadas

- Los estándares ISO definen 5 sentencias agregadas:

**COUNT** retorna el número de valores en una columna específica.

**SUM** Retorna la suma de valores de una columna

**AVG** Retorna el promedio de valores de una columna

**MIN** Retorna el valor más pequeño de una columna

**MAX** Retorna el valor más grande de una columna



Arquitectura Tecnológica de Aplicaciones WEB

465<sup>o</sup>

---

---

---

---

---

---

---

---

## SELECT Sentencias agregadas

- Cada una opera sobre una columna de una tabla y retorna un sólo valor.

- **COUNT**, **MIN**, y **MAX** aplican sobre campos numéricos y no-numéricos, pero **SUM** y **AVG** deben usarse sólo en campos numéricos.

- Además con **COUNT(\*)**, cada función elimina primero los nulos y opera sólo con los valores que quedan no-nulos.



Arquitectura Tecnológica de Aplicaciones WEB

466<sup>o</sup>

---

---

---

---

---

---

---

---

## SELECT Statement - Aggregates

- **COUNT(\*)** cuenta todas las filas de una tabla, sin importar si anula o no los valores duplicados que existan.

- Se puede usar **DISTINCT** antes del nombre de una columna para eliminar los registros duplicados.

- **DISTINCT** no tiene efecto con **MIN/MAX**, pero puede tenerlo con **SUM/AVG**.

- Funciones agregadas Aggregate functions can be used only in **SELECT** list and in **HAVING** clause.



Arquitectura Tecnológica de Aplicaciones WEB

467<sup>o</sup>

---

---

---

---

---

---

---

---



## Ejemplo9 Use of MIN, MAX, AVG

Encontrar el maximo minimo y promedio de los salarios del staff.

```
SELECT MIN(salary) AS min,  
       MAX(salary) AS max,  
       AVG(salary) AS avg  
FROM staff;
```

Table 13.15 Result table for Example 13.15.

count	sum
2	54000.00
(1 row)	



---

---

---

---

---

---

---

---

## INSERT

```
INSERT INTO table_name [ (column_list) ]  
VALUES (data_value_list)
```

- *column\_list* es opcional.
- Si se omite, SQL assume una lista de todas las columnas en su orden original.



---

---

---

---

---

---

---

---

## Ejemplo10 INSERT ... VALUES

```
INSERT INTO staff  
VALUES ('SG16', 'Alan', 'Brown',  
       '67 Endrick Rd, Glasgow G32 8QX',  
       '0141-211-3001', 'Assistant', 'M', '25-May-57',  
       8300, 'WN848391H', 'B3');
```



---

---

---

---

---

---

---

---

## Ejemplo11 INSERT using Defaults

```
INSERT INTO staff (sno, fname, lname, position,  
                  salary, bno)  
VALUES ('SG44', 'Anne', 'Jones', 'Assistant',  
        8100, 'B3');
```



---

---

---

---

---

---

---

## Ejemplo11 INSERT using Defaults

```
INSERT INTO staff  
VALUES ('SG44', 'Anne', 'Jones', NULL, NULL,  
        'Assistant', NULL, NULL, 8100, NULL, 'B3');
```



---

---

---

---

---

---

---

## UPDATE

```
UPDATE table_name  
SET column_name1 = data_value1  
  [, column_name2 = data_value2...]  
[WHERE search_condition]
```

- *table\_name*
- SET clause



---

---

---

---

---

---

---

## UPDATE

- WHERE clause is optional:
  - If omitted, named columns are updated for all rows in table.
  - If specified, only those rows that satisfy *search\_condition* are updated.
- New *data\_value(s)* must be compatible with data type for corresponding column.



---

---

---

---

---

---

---

---

## Ejemplo12 UPDATE All Rows

Give all staff a 3% pay increase.

```
UPDATE staff
SET salary = salary*1.03;
```



---

---

---

---

---

---

---

---

## Ejemplo13 UPDATE Specific Rows

Give all Managers a 5% pay increase.

```
UPDATE staff
SET salary = salary*1.05
WHERE position = 'Manager';
```

- WHERE clause finds rows that contain data for Managers. Update is applied only to these particular rows.



---

---

---

---

---

---

---

---

## Ejemplo14 UPDATE Multiple Columns

Promote David Ford (sno = 'SG14') to Manager and change his salary to 18,000.

```
UPDATE staff
SET position = 'Manager', salary = 18000
WHERE sno = 'SG14';
```



---

---

---

---

---

---

---

---

## DELETE

```
DELETE FROM table_name
[WHERE search_condition]
```

- *table\_name* can be name of a base table or an updatable view.
- *search\_condition* is optional; if omitted, all rows are deleted from table. This does not delete table. If *search\_condition* is specified, only those rows that satisfy condition are deleted.



---

---

---

---

---

---

---

---

## Ejemplo15 DELETE Specific Rows

Delete all viewings that relate to property PG4.

```
DELETE FROM viewing
WHERE pno = 'PG4';
```



---

---

---

---

---

---

---

---

## Ejemplo16 DELETE All Rows

Delete all records from the Viewing table.

```
DELETE FROM viewing;
```



---

---

---

---

---

---

---

---

## ISO SQL Data Types

Table 13.36 ISO SQL data types.

<i>Data type</i>	<i>Declarations</i>			
character	CHAR,	VARCHAR		
bit	BIT,	BIT VARYING		
exact numeric	NUMERIC,	DECIMAL,	INTEGER,	SMALLINT
approximate numeric	FLOAT,	REAL,	DOUBLE PRECISION	
datetime	DATE,	TIME,	TIMESTAMP	
interval	INTERVAL,			



---

---

---

---

---

---

---

---

## CREATE TABLE (Basic)

```
CREATE TABLE table_name  
(col_name data_type [NULL | NOT NULL] [...])
```

- Creates a table with one or more columns of the specified *data\_type*.
- NULL (default) indicates whether column can contain *nulls*.
- With NOT NULL, system rejects any attempt to insert a null in the column.



---

---

---

---

---

---

---

---

## CREATE TABLE (Basic)

- Primary keys should always be specified as NOT NULL.
- Foreign keys are often (but not always) candidates for NOT NULL.



---

---

---

---

---

---

---

## Ejemplo17 CREATE TABLE

```
CREATE TABLE staff(  
    sno          VARCHAR(5)          NOT NULL,  
    fname        VARCHAR(15)         NOT NULL,  
    lname        VARCHAR(15)         NOT NULL,  
    address      VARCHAR(50),  
    tel_no       VARCHAR(13),  
    position     VARCHAR(10)         NOT NULL,  
    sex          CHAR,  
    dob          DATETIME,  
    salary       DECIMAL(7,2)        NOT NULL,  
    nin          CHAR(9),  
    bno          VARCHAR(3)          NOT NULL);
```



---

---

---

---

---

---

---

## Ejemplo18 CREATE TABLE

```
CREATE TABLE property_for_rent(  
    pno          VARCHAR(5)          NOT NULL,  
    street       VARCHAR(25)         NOT NULL,  
    area         VARCHAR(15),  
    city         VARCHAR(15)         NOT NULL,  
    pcode        VARCHAR(8),  
    type         CHAR(1)            NOT NULL,  
    rooms        SMALLINT            NOT NULL,  
    rent         DECIMAL(6,2)        NOT NULL,  
    ono          VARCHAR(5)          NOT NULL,  
    sno          VARCHAR(5),  
    bno          VARCHAR(3)          NOT NULL);
```



---

---

---

---

---

---

---

## DROP TABLE

**DROP TABLE** tbl\_name [RESTRICT | CASCADE]

e.g. **DROP TABLE** property\_for\_rent;

- Removes named table and all rows within it.
- With **RESTRICT**, if any other objects depend for their existence on continued existence of this table, SQL does not allow request.
- With **CASCADE**, SQL drops all dependent objects (and objects dependent on these objects).



---

---

---

---

---

---

---

---