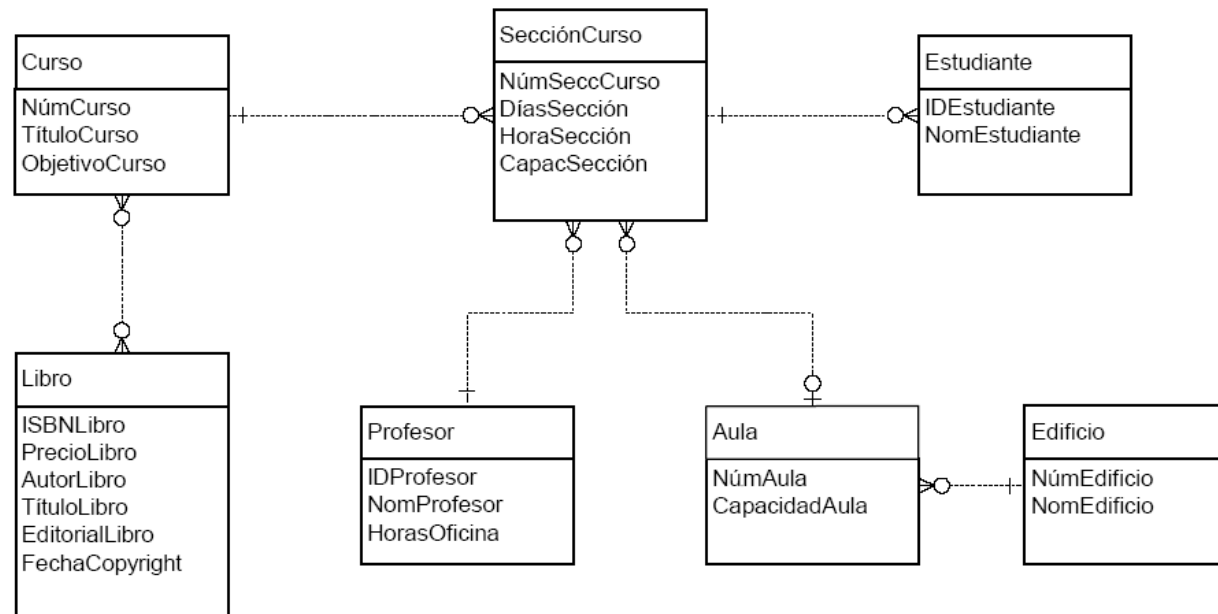


SUPORT INFORMÀTIC PER LA LOGÍSTICA

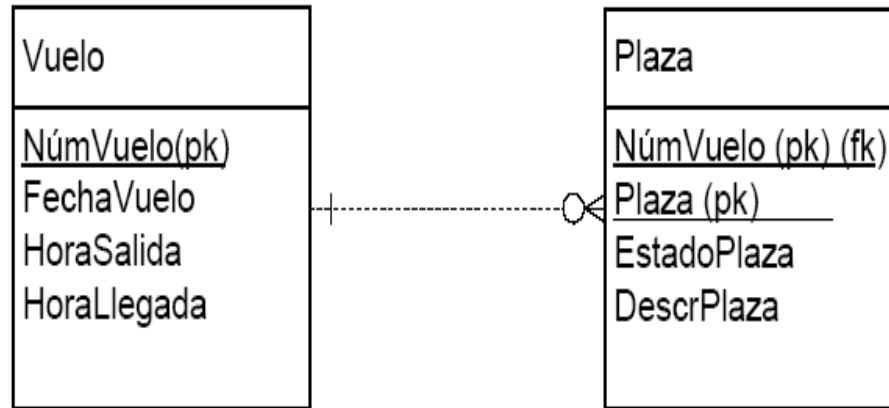
BASE DE DADES

Conceptes Entitat-Relació (ER)

- Entitats (objectes, taules)
- Atributs (propietats)
- Claus (primàries i externes, forànies)
- Relacions
- Integritat referencial



Model ER: Claus i Relacions



Relació 1:N (cas general de 1:1)

Un vol té fins a N places, i una plaça només pot estar associada a un vol.

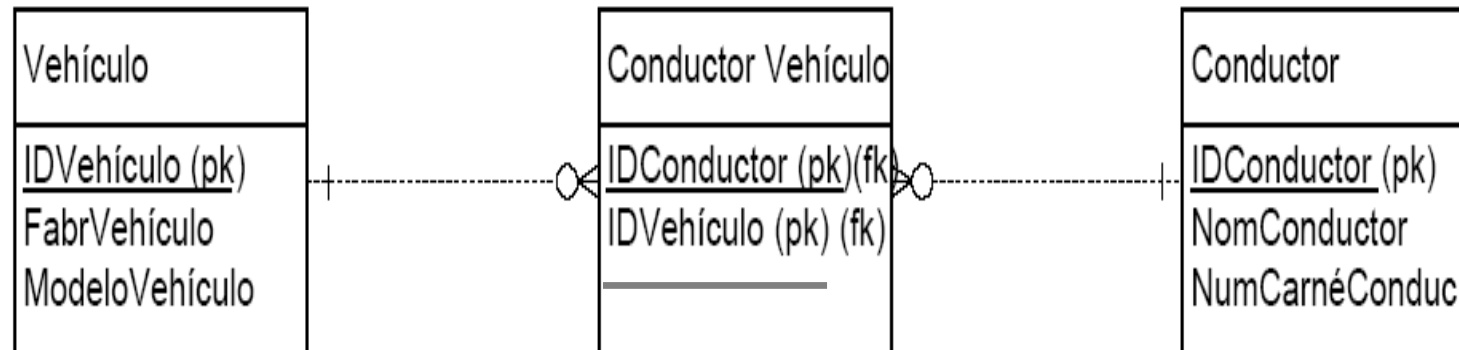
Vuelo			
NúmVuelo	FechaVuelo	HoraSalida	HoraLlegada
243	9/24/00	9:00am	11:00am
253	9/24/00	10:00am	12:30pm
52	9/24/00	11:00am	2:00pm

Plaza			
NúmVuelo	NumPlaza	EstadoPlaza	DescrPlaza
243	8A	Confirmada	Ventanilla
243	7D	Reservada	Pasillo
243	14E	Abierta	Central
253	1F	Abierta	Ventanilla
253	43A	Confirmada	Ventanilla

Model ER: Claus i Relacions (II)

Relació N:N

Un vehicle té pot estar conduit per molts conductors, i un conductor pot conduir diferents vehicles.

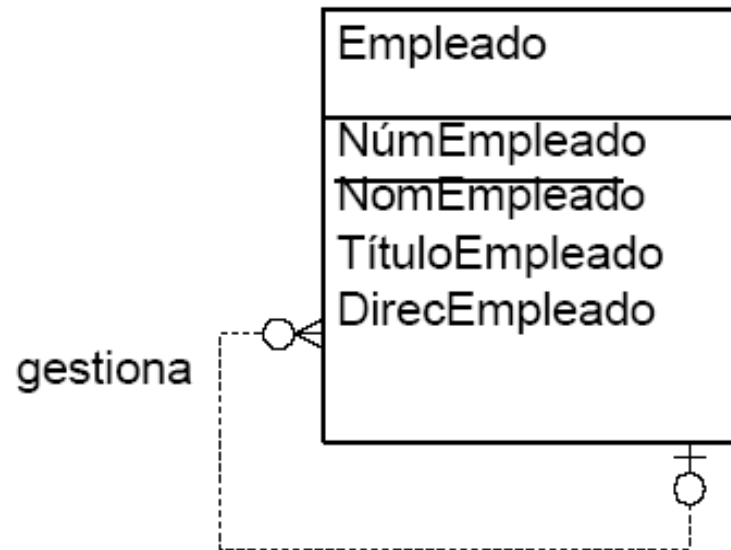


Vehículo		
IDVehículo	FabrVehículo	ModeloVehic
35	Volvo	Wagon
33	Ford	Sedan
89	GMC	Camión

Conduc Vehículo	
IDVehículo	IDCond
35	900
35	253
89	900

Conductor		
IDCond	NomConduc	NúmCarnéConduc
253	Ken	A23423
900	Jen	B89987

Model ER: Claus i Relacions (III)



Relacions recurrents.

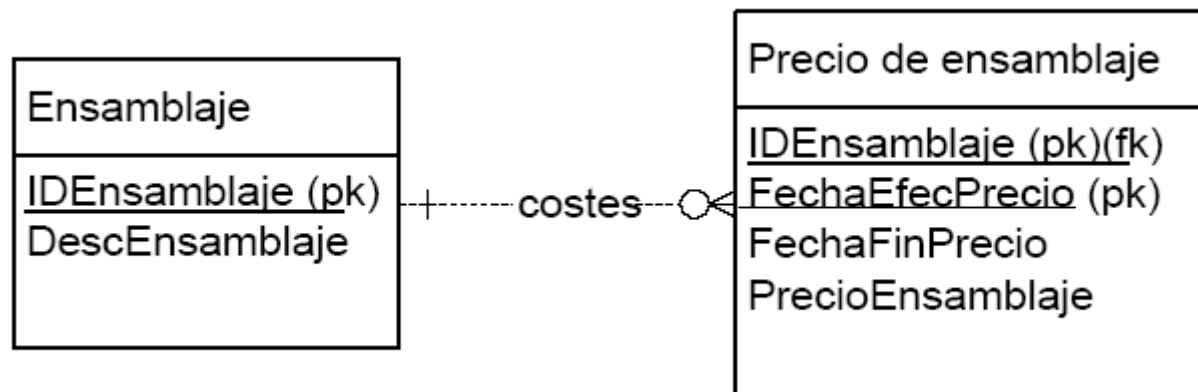
Un empleat té un responsable. Un Responsable pot tenir fins a N subordinats.

Empleado(
NumEmpleado (pk),
NomEmpleado,
Título Empleado,
DirecEmpleado,
ResponsableId (fk))

Model ER: Claus i Relacions (IV)

Relacions temporals:

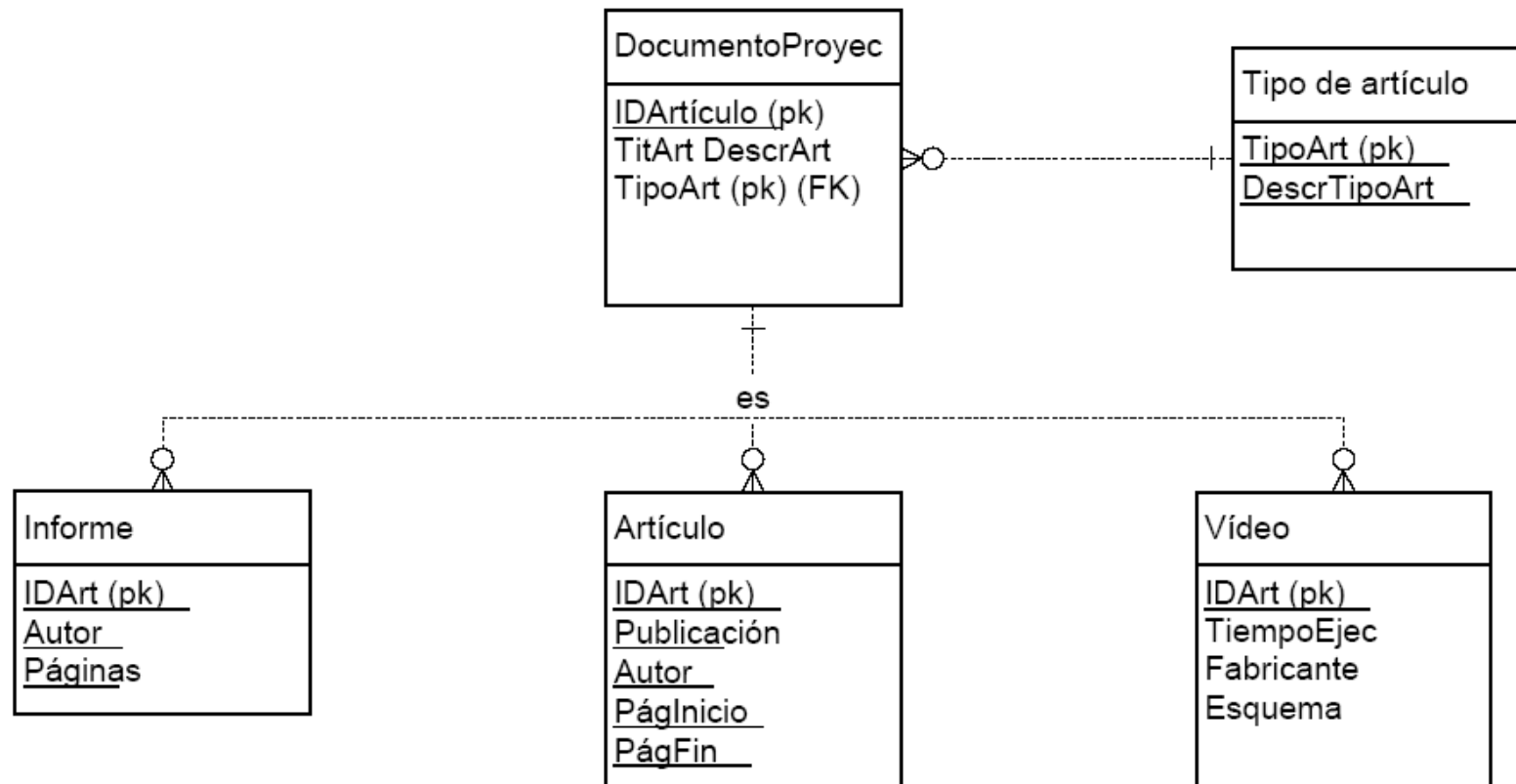
El ensamblatge té uns costos que depen del preu dels components en el moment d'ensamblatge.



Model ER: Claus i Relacions (V)

Jerarquia (Herència)

Un document de projecte pot ser un informe, article o video.



Model ER: Integritat Refencial

- **Manté la consistència de les claus externes**
 - Cada clau forania coincideix amb una clau principal o és null.
- Regles:
 - **Restrict:** No és permet l'eliminació de les files que contenen claus principals si les files que contenen les claus foranies no s'han eliminat previament.
 - P.ex: al eliminar un departament cal eliminar els empleats que depenen del departament.
 - **Cascada:** És produeixen modificacions en cascada.
 - Eliminació
 - P.ex: al elimnar una comanda, s'eliminen els articles
 - Actualització.
 - P.ex: al canviar la clau del departament, es cambien la referència al departament dels empleats d'aquest.

Normalització

- **1era Forma Normal**
 - Totes les files han de tenir una longitud fixa. No es poden posar llistes de longitud variable en una fila.
- **2ona Forma Normal**
 - Tots els camps que no son clau han de ser funció de la clau

Pieza	Almacen	Cantidad	DirAlmacen
42	Boston	2000	24 Main St
333	Boston	1000	24 Main St
390	New York	3000	99 Broad St

No normalitzat

Pieza	Almacen	Cantidad
42	Boston	2000
333	Boston	1000
390	New York	3000

Almacen	DirAlmacen
Boston	24 Main St
New York	99 Broad St

Solució

Normalització (II)

- **3era Forma Normal**

- Els camps que no són claus no poden ser funció d'una camp que no sigui clau.

Empleado	Departamento	UbicacionDepart
234	Finance	Boston
223	Finance	Boston
399	Operations	Washington

No normalitzat

Empleado	Departamento
234	Finance
223	Finance
399	Operations

Departamento	UbicacionDepart
Finance	Boston
Operations	Washington

Solució

Normalització (III)

- **4rta Forma Normal**

- Una fila no hauria de contenir només un atribut multievaluat.

Empleado	Destreza	Idioma
Brown	cook	
Brown	type	
Brown		French
Brown		German
Brown		Greek

No normalitzat

Empleado	Destreza
Brown	cook
Brown	type

Empleado	Idioma
Smith	French
Smith	German
Smith	Greek

Solució

Normalització (IV)

- **5ena Forma Normal**

- No es pot construir una registre a partir de la composició de registres més petits.
- Exemple:
 - Els agents representen companyies
 - Les companyies elaboren productes
 - Els agents venen productes

Agente	Compañia	Producto
Smith	Ford	car
Smith	Ford	truck
Smith	GM	car
Smith	GM	truck
Jones	Ford	car
Jones	Ford	truck
Brown	Ford	car
Brown	GM	car
Brown	Toyota	car
Brown	Toyota	bus

No normalitzat

Agente	Compañia
Smith	Ford
Smith	GM
Jones	Ford
Brown	Ford
Brown	GM
Brown	Toyota

Compañia	Producto
Ford	car
Ford	truck
GM	car
GM	truck
Toyota	car
Toyota	bus

Agente	Producto
Smith	car
Smith	truck
Jones	car
Jones	truck
Brown	car
Brown	bus

Solució

SQL: Standard Query Language

- SQL s'utilitza per:
 - Definició de dades, taules i vistes (taules virtuals)
 - Recuperació de dades (desde l'usuari, o programa)
 - Manipulació de dades
 - Control d'accés
 - Compartició de dades (usuaris concurrents)
 - Mantenir l'integrat de les dades
- SQL no és:
 - Un llenguatge de programació complert (aprox. 30 instruccions)
- SQL és:
 - Un estàndard de subllenguatge, que va insertat en un altre llenguatge (java, c, delphi, php), o a una utilitat d'accés a BD.
 - Perquè es un estàndard és multi-plataforma, encara que té problemes de portabilitat.

SQL: Portabilitat

- Encara que SQL hauria de ser estàndard (SQL92), diferents implementacions tenen aspectes que varien:
 - Codis d'error
 - Tipus de dades que suporten (moneda, strings, dates, blobs)
 - Taules del sistema
 - SQL interactiu
 - SQL dinàmic, Stored-procedures, triggers
 - Interfície de programació

Diferents SGBD (Sistemes Gestors de Base de Dades) com:

Oracle, Informix, DB2, MySQL, SQLServer

Tenen petites diferències que dificulten la portabilitat. Tanmateix, el nucli del SQL92 es compartit per la gran majoria d'implementacions.

SQL: Esquema de DB

L'esquema és l'estructura de la DB, les dues instruccions principals són:
CREATE TABLE i DROP TABLE. Hi ha diferències entre SGBD's, sobretot en els tipus de dades.

Sintaxis:

```
CREATE TABLE nom_taula (  
    nom_columna tipus_de_dades [NULL, NOT NULL, UNIQUE]  
        [auto_increment],  
  
    ...,  
    [PRIMARY KEY (nom_columna),]  
    [UNIQUE (nom_columna),]  
    [FOREIGN KEY nom_columna REFERENCE nom_taula(nom_columna),]  
    [CONSTRAINT nom_condicio CHECK condicio]  
)
```

```
DROP TABLE nom_taula
```

SQL: Esquema de DB (II)

Exemples:

CREATE TABLE Client (

```
id          int          NOT NULL AUTO_INCREMENT,  
dni         char(10)      NOT NULL,  
nom        char(100)   NOT NULL,  
descripcio text        NULL,  
idRepr     int          NOT NULL,  
alta       date       NOT NULL,  
PRIMARY KEY (id),  
UNIQUE (dni),  
FOREIGN KEY idRepr REFERENCE Representat (id),  
CONSTRAINT constr1 CHECK (alta>'2002/01/01')
```

);

DROP TABLE Client;

SQL: SELECT

- Recuperació d'informació, l'instrucció més important.
- Sintaxis:

SELECT columna

FROM taula o vista

[**INTO** nova taula]

WHERE relació o files específiques (restriccions)

[**GROUP BY** condicions d'agrupació (columnes)]

[**HAVING** propietats de grup (files)]

[**ORDER BY** criteris d'ordenació]

SQL: Exemples de SELECT

- Seleccionar totes les columnes:
 - **SELECT * FROM** Oficinas;
- Eliminar duplicats:
 - **SELECT DISTINCT** Region **FROM** Oficinas;
- Operacions sobre columnes: AVG, MAX, MIN, COUNT, SUM
 - **SELECT AVG**(Import) **FROM** Pedidos **WHERE** Clien=221;
 - **SELECT COUNT**(*) **FROM** Oficinas **WHERE** Region='West';
- Comprovació del nulls:
 - **SELECT** Nombre **FROM** RepVentas **WHERE** Cupo **IS NULL**;
- Operadors del select: (SELECT * FROM taula...
 - WHERE Dto*Importe > 5000. (<, >, >=, <=, =)
 - WHERE ((Cupo >= 50000) AND (Cupo <= 100000)) (**AND, OR, NOT**)
 - WHERE Cupo **BETWEEN** 50000 **AND** 100000;
 - WHERE Estado **IN** ('CO','UT','TX');
 - WHERE Telf **LIKE** '21%' (**LIKE, NOT LIKE, _ %**)

SQL: INSERT i DELETE

- INSERT: s'utilitza per insertar noves files
 - Sintaxis:
INSERT INTO taula [(columnes)]
VALUES (valors dels atributs columnes);
 - Exemple:
INSERT INTO Oficinas **VALUES** ('55','Dallas','West',200000,0);
- DELETE: s'utilitza per esborrar files
 - Sintaxis:
DELETE FROM taula
WHERE condició
 - Exemple:
DELETE FROM Clientes **WHERE** Empresa = 'Connor Co';
Compte amb inconsistències derivades: cascade, restrict.

SQL: UPDATE

- S'utilitza per modificar el valor dels atributs
- Sintaxis:
UPDATE taula
SET columna = nou valor
WHERE condició
- Exemple:
 - Augmentar un 30% el preu de les excavadores en les comandes:
UPDATE Pedidos **SET** Importe=Importe*1.3
WHERE Prod='Excavadora'

SQL SELECT: COUNT i GROUP BY

Tenim la taula **Piezas**:

PiezaID	Distribuidor
123	A
234	A
345	B
362	A
2345	C
3464	A
4533	C

- COUNT: s'utilitza per comptar el numero de files.

– Exemple:

```
SELECT COUNT(*) FROM Piezas WHERE  
Distribuidor='A'
```

Resultat: 4

- GROUP BY: s'utilitza per agrupar per un atribut. No té perquè anar amb el count.

– Exemple:

```
SELECT Distribuidor, COUNT(*) AS  
CuentaPiezas FROM Piezas GROUP BY  
Distribuidor;
```

Resultat:

<i>Distribuidor</i>	<i>CuentaPiezas</i>
A	4
B	1
C	2

El GROUP BY sempre per una columna del SELECT!

SQL: Relacions

- Quan volem fer un SELECT per mes d'una taula cal trobar una columna de relació. Una bona elecció és:
 - Columnes que siguin un camp clau (primària o forània)
 - Les columnes de relació han de ser tipus de dades compatibles.
 - Les columnes que contenen NULL's no son bones candidates.
- Exemple:
 - Obtenir un llistat de les comandes de mes de 25k dòlars, en el que apareixi el nom del venedor i del client.

```
SELECT NumPedido, Importe, Empresa, Nombre
FROM Pedidos, RepVentas
WHERE Clien = NumClien AND RepClien = NumRep
AND Importe >= 25000;
```

Resultat:

<i>NumPedido</i>	<i>Importe</i>	<i>Empresa</i>	<i>Nombre</i>
1	34000	Connor Co	Jen Jones
3	500000	AmaratungaEnterprise	Jen Jones

SQL: Comentaris sobre relacions

- Cal tenir en compte amb el * a les relacions
- Si un camp té el mateix nom en les taules cal qualificar el nom amb el nom de la taula:
taula1.nom_camp, taula2.nom_camp
SELECT Client.id, Producte.id ...
- Relacions dins d'una mateixa taula (exemple, taula Empleados)

NumEmp	Nombre	Posicion	Enc
105	Mary Smith	Analista	104
104	Sally Silver	Analista Senior	101
101	Eileen Howe	Encarregat	

Llista d'empleats i dels seus encarregats:

NO! SELECT Nombre, Nombre FROM Empleados, Empleados WHERE Enc=NumEmp;

Utilització d'alias: SELECT Emp.Nombre, Enc.Nombre FROM Empleados Emp, Empleados Enc WHERE Emp.Enc = Emp.NumEmp;

SQL: Subconsultes

- Les subconsultes de SQL permeten utilitzar els resultats d'una consulta com a part d'una altra consulta.
 - Permeten dividir una consulta en parts i reconstruir-la (disseny descendent)
 - Permeten realitzar consultes que no seria possible sense subconsultes. (p.ex: mysql no suporta sub-consultes)
 - A vegades és la forma natural d'escriure una consulta
- Consideracions:
 - Sempre figuren en una clàusula WHERE (o HAVING)
 - Només poden retornar resultats d'una columna
 - El Select només pot tenir un camp
 - No permeten un ORDER BY, no tindria sentit.

SQL: Subconsultes (II)

- **Exemple:**

Obtenir un llistat de les ciutats de les oficines que el seu cupó de venda sigui superior a la suma dels cupos individuals dels venedors:

1. **SELECT** Ciudad **FROM** Oficinas **WHERE** Objetivo > ???
2. ??? És la suma dels cupos dels venedors
3. Que traduït al SQL: **SELECT SUM(Cupo) FROM RepVentas WHERE OfiRep=NumOfi;**

Combinades:

SELECT Ciudad **FROM** Oficinas **WHERE** Objetivo >
(SELECT SUM(Cupo) FROM RepVentas WHERE OfiRep=NumOfi)

SQL: Vistes (Views)

- Son taules virtuals que mostren dades als usuaris.
- No són còpies de les dades, fan referència a les dades de les taules subjacents.
- Les BD emmagatzemen la definició de les vistes. Les dades s'actualitzen quan les taules subjacents s'actualitzen.
- Avantatges:
 - Les consultes son més simples pels usuaris.
 - Seguretat: només es permet accedir a les dades de les vistes.
 - Independència: evita que els usuaris o programes modifiquin les taules subjacents.
- Exemple:

CREATE VIEW PedidosClientes AS

```
SELECT NumClien, Empresa, Nombre, NumPedido, Prod, Cant,  
Importe FROM Clientes, RepVentas, Pedidos WHERE  
RepClien=NumRep AND NumClien=Clien
```

SQL: Índexs

- Un índex és un objecte de dades independent que ordena les files per facilitar una recuperació ràpida.
 - Les claus primàries i secundàries s'indexen automàticament.
 - Es poden utilitzar 3 o 4 índexs per taula.
 - Els índexs milloren el rendiment a les lectures (recuperacions, SELECT), però penalitzen les operacions de modificació, inserció i eliminació (UPDATE, INSERT, DELETE)
 - Tradeoff entre consultes (+índexs) i actualitzacions (-índexs)
- Exemple:
 - En una taula Clients, el id és la clau primària, però també es poden realitzar consultes per nom, direcció i codi postal. Id està indexat automàticament. Val la pena indexar la resta d'atributs? Depèn...

Seguretat

- Opcions de seguretat:
 - Utilitzar un username/password del sistema operatiu
 - Utilitzar un username/password per la base de dades
 - Seguretat a nivell d'aplicació (comprova si l'usuari té els privilegis per accedir a una aplicació)
 - Seguretat a nivell de xarxa (Utilització d'una infraestructura de clau pública (PKI), Entitats de certificació, Firma digital)
- Llei de protecció de dades (*Ley Orgánica 15/1999*)
- Tipus d'usuari:
 - DBA (Administrador de la BD)
 - Usuari

Assignació de privilegis sobre la DB, responsabilitat del DBA.

GRANT i REVOKE

GRANT ALL ON NomTaula TO PUBLIC WITH GRANT OPTION

Transaccions

- Una transacció és un grup d'operacions que s'han de tractar com una **unitat atòmica**.
 - **Evitar problemes amb base de dades multiusuaris** (concurrents)
 - Problema de les actualitzacions perdudes: (**sincronització**)
 - Hi ha 7 peces a l'inventari
 - Els processos x i y llegeixen el registre que és 7.
 - x acaba i afegeix 3 peces a l'inventari, escriurà 10.
 - y acaba després i resta 5 peces, escriurà 2.
 - Inconsistència!
 - **Tractament en blocs:**
 - Típics dels processos batch
 - Cal esborrar els subconjunt de clients Z.
 - Falla uns dels DELETES d'un client x.
 - Cal fer el insert de tots els clients ja esborrats per recuperar l'estat inicial?

Transaccions (II)

- Sintaxis:

START TRANSACTION

<SQL>

Si hi ha un error **ROLLBACK**
sinó **COMMIT**

Malauradament, la sintaxi varia entre diferents SGBD:

Per exemple en MySQL:

```
START TRANSACTION;
```

```
SELECT @A:=SUM(salary) FROM table1 WHERE type=1;
```

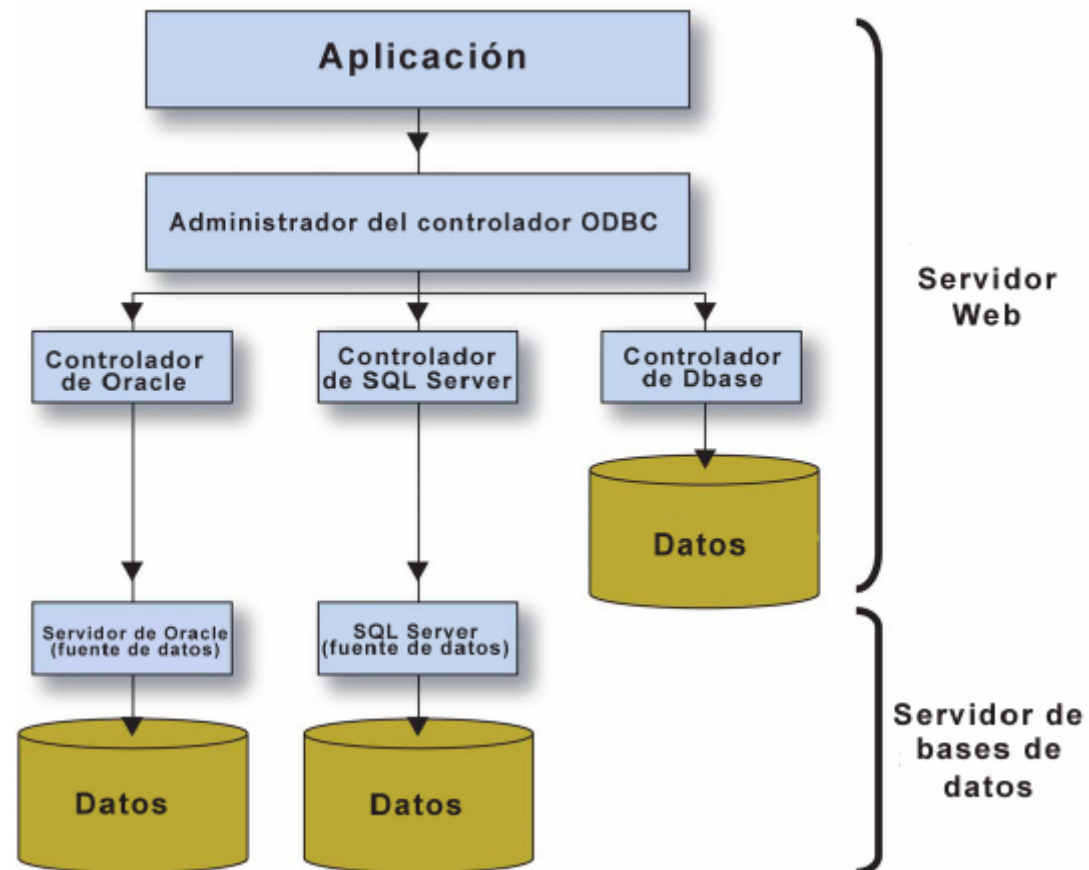
```
UPDATE table2 SET summary=@A WHERE type=1;
```

```
COMMIT;
```

ODBC: Open DataBase Connection

- Cronologia:
 - **Etape inicial:** SQL incrustat (ESQL) en cada servidor de bases de dades (<90).
 - Compilat en aplicacions de servidors, no pot ser modificat pel usuari final.
 - **Segona etapa:** aplicacions clients servidor (mitjans-finals 90):
 - Cada distribuïdor ofereix un API (interfície de programació). Programes clients realitzen consultes a les bases de dades. Cada distribuïdor (SGBD) tenia la seva pròpia API.
 - **Tercera etapa:** ODBC (finals 90 fins ara).
 - API comú i estàndard, que permet accedir a la majoria de bases de dades:
 - Oracle, SQL Server, Sybase, DB2, Informix, MySQL.
 - JDBC és l'equivalent per l'entorn JAVA.

ODBC: Arquitectura



ODBC: Exemple en JAVA-JDBC

```
import java.sql.*;
import twz1.jdbc.mysql.*;
public class TestQuery {
    public TestQuery () { }
    public static void main(String args[]) {
        String url="jdbc:z1MySQL://luna.oit.unc.edu/CES?user=alexadmin";
        Connection con;
        String query = "SELECT * FROM alex_course";
        Statement stmt;
        try {
            Class.forName("twz1.jdbc.mysql.jdbcMysqlDriver");
        } catch (java.lang.ClassNotFoundException e) { System.err.print("ClassNotFoundException: ");
            System.err.println(e.getMessage()); }
        try {
            System.out.println("Trying to connect...");
            con = DriverManager.getConnection (url, "alexadmin", "xxxxxxx");
            System.out.println("connected!");
            stmt = con.createStatement();
            ResultSet result = stmt.executeQuery(query);
            while (result.next()) {
                String name = result.getString(1) + " " + result.getString(2);
                System.out.println(name);
            }
            stmt.close();
            con.close();
        } catch(SQLException ex) {
            System.err.print("SQLException: ");
            System.err.println(ex.getMessage()); }
    }
}
```

Cadena de connexió

La sentència SQL

Càrrega del driver JDBC

Obra una connexió

Crea un statement

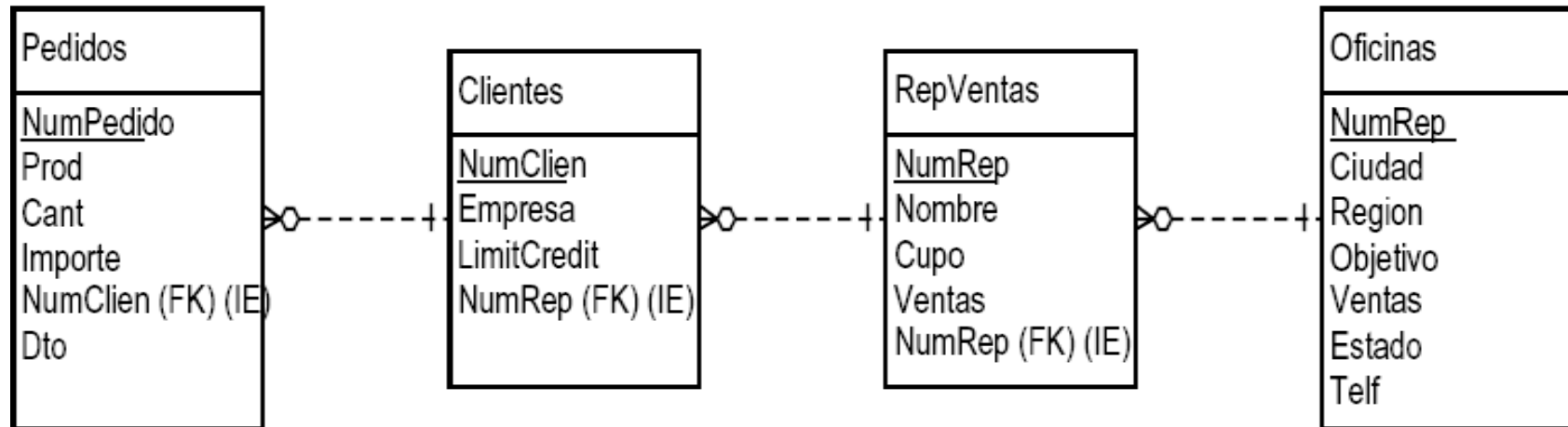
Obté un ResultSet

Recorregut sobre el ResultSet (les files del resultat d'una consulta)

SQL: Exercicis

1. Quantes oficines de ventes hi ha a la regió West?
2. I que a més tinguin unes ventes superiors a 100000.
3. Quin es el límit mitjà de crèdit dels clients amb un límit de crèdit inferior a 1000000?
4. Obtenir un llistat dels noms i números de telèfon dels representats.
5. Mostrar tots els clients amb comandes o límits de crèdit més grans que 50000 dòlars. Llistar també el crèdit i el 'cupo'.
6. L'oficina de TX s'ha mogut a Las Vegas, estat de Nevada (NE)
7. Eliminar les oficines que tenen perdues.

Annex: Exemple d'esquema



Annex: Exemple de Taules

Pedidos

NumPedido	Clien	Prod	Cant	Importe	Dto
1	211	Excavadora	7	31000.00	0.2
2	522	Remachadora	2	4000.00	0.3
3	522	Grua	1	500000.00	0.4

Cientes

NumClien	Empresa	RepClien	LimitCredit
211	Connor Co	89	50000.00
522	AmaratungaEnterprise	89	40000.00
890	Feni Fabricators	53	1000000.00

RepVentas

NumRep	Nombre	OfiRep	Cupo	Ventas
53	Bill Smith	1	100000.00	0.00
89	Jen Jones	2	50000.00	130000.00

Oficinas

NumOfi	Ciudad	Estado	Region	Objetivo	Ventas	Telf
1	Denver	CO	West	3000000.00	130000.00	970586
2	New York	NY	East	200000.00	300000.00	212942
57	Dallas	TX	West	0.00	0.00	214781

Annex: Scripts de la BD

```
CREATE TABLE Oficinas (  
    NumOfi INT NOT NULL,  
    Ciudad CHAR(100) NOT NULL,  
    Estado CHAR(2) NOT NULL,  
    Region CHAR(10) NOT NULL,  
    Objetivo DECIMAL,  
    Ventas DECIMAL,  
    Telf CHAR(10),  
    PRIMARY KEY(NumOfi)  
);  
  
CREATE TABLE RepVentas (  
    NumRep INT NOT NULL,  
    Nombre VARCHAR(100) NOT NULL,  
    OfiRep INT NOT NULL,  
    Cupo DECIMAL,  
    Ventas DECIMAL,  
    PRIMARY KEY(NumRep),  
    FOREIGN KEY (OfiRep) REFERENCES Oficinas(NumOfi)  
);  
  
CREATE TABLE Clientes (  
    NumClien INT NOT NULL,  
    Empresa CHAR(50) NOT NULL,  
    RepClien INT NOT NULL,  
    LimitCredit DECIMAL,  
    PRIMARY KEY (NumClien),  
    FOREIGN KEY (RepClient) REFERENCES RepVentas(NumRep)  
);  
  
CREATE TABLE Pedidos (  
    NumPedido INT NOT NULL,  
    Clien INT NOT NULL,  
    Prod CHAR(20),  
    Cant INT NOT NULL,  
    Import DECIMAL,  
    Dto DECIMAL,  
    PRIMARY KEY(NumPedido),  
    FOREIGN KEY (Clien) REFERENCES Clientes(NumClien)  
);
```

```
DROP TABLE Pedidos;  
DROP TABLE Clientes;  
DROP TABLE RepVentas;  
DROP TABLE Oficinas;
```

```
INSERT INTO Pedidos VALUES (1,211,'Excavadora',7,31000.00,0.2);  
INSERT INTO Pedidos VALUES (2,522,'Remachadora',2,4000.00,0.3);  
INSERT INTO Pedidos VALUES (3,522,'Grua',1,500000.00,0.4);  
  
INSERT INTO Clientes VALUES (211,'Connor Co',89,50000.00);  
INSERT INTO Clientes VALUES (522,'AmaratungaEnterprise',89,40000.00);  
INSERT INTO Clientes VALUES (890,'Feni Fabricators',53,1000000.00);  
  
INSERT INTO RepVentas VALUES (53,'Bill Smith',1,100000.00,0.00);  
INSERT INTO RepVentas VALUES (89,'Jen Jones',2,50000.00,130000.00);  
  
INSERT INTO Oficinas VALUES  
(1,'Denver','CO','West',3000000.00,130000.00,'970586');  
INSERT INTO Oficinas VALUES (2,'New  
York','NY','East',200000.00,300000.00,'212942');  
INSERT INTO Oficinas VALUES (57,'Dallas','TX','West',0.00,0.00,214781);
```

Annex: Referències

- **Material:**
 - Presentació: <http://www.lsi.upc.es/~jmpujol/gsel/gsel-db.zip>
 - Scripts de la BD:
 - Create <http://www.lsi.upc.es/~jmpujol/gsel/gsel-create-db.sql>
 - Delete <http://www.lsi.upc.es/~jmpujol/gsel/gsel-drop-db.sql>
 - Inserts <http://www.lsi.upc.es/~jmpujol/gsel/gsel-insert-db.sql>
- **Mysql**
 - Servidor: <http://dev.mysql.com/downloads/>
 - Client gràfic: <http://dev.mysql.com/downloads/query-browser/>