

# Tema 3: Introducció a SQL

## Bases de Dades

Marta Tarrés-Puertas

Enginyeria de Sistemes TIC  
Universitat Politècnica de Catalunya  
<http://itic.cat>

25 de setembre de 2013

## 1 Introducció a SQL

## 2 Creació i Modificació de Bases de Dades

- Definició de Dades
- Creació de taules
- Eliminació i modificació de taules

## 3 Consultes

- Condicions de cerca simples
- Predicat Like
- Predicat in/not in
- Valors null
- Presentació de resultats: Ordenació de tuples
- Operacions de conjunts
- Funcions d'agregació
- La clàusula Having
- Subconsultes niuades

## 4 Més aspectes de la modificació d'una BD

- IBM Sequel language desenvolupat com a part del projecte System R en a IBM San Jose Research Laboratory
- Anomenat posteriorment Structured Query Language (SQL)
- ANSI i ISO standard SQL
  - SQL-86, SQL-89, SQL-92
  - SQL:1999, SQL:2003, SQL:2008

- Taules = Relacions
- Columnes = Atributs
- Files = Tuples

usuaris(email,nom,cognom,poblacio,dataNaixement,pwd)

amistats(email1,email2,estat)

Entitat Usuari = Taula Users

email	nom	cognom	poblacio	dataNaixement	pwd
johnQ@themail.cat	John	Quimet	Manresa	1978-12-25	johnQ2578
sarahC@themail.cat	Sarah	Hidalgo	Barcelona	1984-10-4	sarah22
annaL@themail.cat	Anna	Lleva	Manresa	1990-11-9	abracadabra
pereG@mail.com	Pere	Guzman	València	1970-3-24	24031970
lluisT@mail.com	Lluis	Talavera	Madrid	1972-7-8	08071972

Relació d'amistat reflexiva entre usuaris

email1	email2	estat
johnQ@themail.cat	sarahC@themail.cat	Aprovada
sarahC@themail.cat	pereG@mail.com	Aprovada
annaL@themail.cat	sarahC@themail.cat	Pendent
pereG@mail.com	johnQ@themail.cat	Rebutjada
lluisT@mail.com	johnQ@themail.cat	Rebutjada

- L'esquema de cada relació
- Els valors de domini associats amb cada atribut.
- Restriccions d'integritat

**create** database nom

**drop** database nom

**create** schema nom authorization usuari

**drop** schema nom **restrict|cascade**

# Create table

```
create table r (A1 D1, A2 D2, ..., An Dn,  
(restricció_taula_1),  
...,  
(restricció_taula_k))
```

- *r* nom de la relació (nom de la taula)
- cada *Ai* és un nom d'atribut de l'esquema de la relació *r* (definició de columna)  
nom\_columna (tipus\_dades|domini) [def\_defecte][restriccions\_columna]
- *Di* és el valor dels tipus de dades en el domini de l'atribut *Ai*
- restriccions taula: regles d'integritat

```
create table instructor  
(ID char(5),  
name varchar(20) not null,  
dept_name varchar(20),  
salary numeric(8,2));
```

```
insert into instructor values ('10211', 'Smith', 'Biology', 66000);  
insert into instructor values ('10211', null, 'Biology', 66000);
```



- char(longitud)
- varchar(longitud)
- int
- smallint
- numeric(digitsNombres,digitsDecimals)
- real, double precision
- float(n)
- date. Ex.: '1978-12-25'
- timestamp. Ex.: '1998-7-8 9:30:05'

# nou Domini en SQL

```
create domain dom_ciuatats as char(20)  
  constraint ciuatats_valides  
  check (value in ('Barcelona','Tarragona','Lleida','Girona'));
```

només s'esborra si no s'usa enlloc

```
drop domain nom_domini [restrict]cascade]  
drop domain dom_ciuatats restrict;
```

Eliminar restricció antiga

```
alter domain dom_ciuatats drop constraint ciuatats_valides;  
alter domain dom_ciuatats add constraint ciuatats_valides  
  check (value in ('Barcelona','Tarragona','Lleida','Girona','Mataro'));
```

Definició de valor per defecte en columna

telefon **integer default null**

data **date default current\_date**

# Restriccions d'integritat en el create table

- not null
- primary key (A1, ..., An ): automàticament not null
- foreign key (Am, ..., An ) references r
- unique el conjunt de les columnes especificades no pot tenir repetits. Clau alternativa
- check la taula ha de complir les condicions especificades

```
create table instructor (  
  ID char(5),  
  number_ss numeric(6) not null,  
  name varchar(20) not null,  
  dept_name varchar(20),  
  data_inici date,  
  data_fi date default null,  
  salary real check (salary > 7000),  
  primary key (ID),  
  unique (number_ss),  
  foreign key (dept_name) references department)  
  check data_inici < data_fi;
```

# Exemple mini-xarxa social

usuaris(email,nom,cognom,poblacio,dataNaixement,pwd)  
amistats(email1,email2,estat)

```
create table usuaris (  
  email varchar(20),  
  nom varchar(20) ,  
  cognom varchar(20) ,  
  poblacio varchar(20),  
  dataNaixement date,  
  pwd varchar(20),  
  primary key (email)  
);
```

```
create domain estatsAmistat as varchar(10)  
constraint estatsValids  
check (value in( 'Aprovada', 'Rebutjada', 'Pendent'));  
create table amistats (  
  email1 varchar(20),  
  email2 varchar(20) ,  
  estat estatsAmistat,  
  primary key(email1,email2),  
  foreign key(email1) references usuaris,  
  foreign key(email2) references usuaris  
);
```

**foreign key** (nom) references taula  
[**on delete**|**no action**|**cascade**|**set default**|**set null**]  
[**on update**|**no action**|**cascade**|**set default**|**set null**];

- no action = restricció
- cascade = actualització en cascada
- set null = anul·lació
- set default = variació de set null, on en lloc de posar valors nuls es pot posar el valor especificat per defecte

instructor(ID,number-ss,name,dept-name,data-inici,data-fi,salary)  
student(ID,name,dept-name,tot-cred)  
takes(ID,course-id,sec-id,semester,year,grade)  
section(course-id,sec-id,semester,year,building,room-nbr)  
couse(course-id,title, dept-name,credits)  
department(dept-name,building)

# Creació de les taules BD university

```
create table student (  
  ID varchar(5) primary key,  
  name varchar(20) not null,  
  dept_name varchar(20),  
  tot_cred numeric(3,0),  
  foreign key (dept_name) references department );
```

```
create table takes (  
  ID varchar(5),  
  course_id varchar(8),  
  sec_id varchar(8),  
  semester varchar(6),  
  year numeric(4,0),  
  grade varchar(2),  
  primary key (ID, course_id, sec_id, semester, year),  
  foreign key (ID) references student,  
  foreign key (course_id, sec_id, semester, year) references section );
```



# Creació de les taules BD university

```
create table course (  
  course_id varchar(8) primary key,  
  title varchar(50),  
  dept_name varchar(20),  
  credits numeric(2,0),  
  foreign key (dept_name) references department );
```

```
insert into course values ('CS-437', 'Database Systems', 'Comp. Sci.', 4);
```

```
insert into course (course_id, title, dept_name, credits)  
  values ('CS-437', 'Database Systems', 'Comp. Sci.', 4);
```

```
insert into student values ('3003', 'Green', 'Finance', null);
```

```
insert into usuaris (email,nom,cognom,poblacio,dataNaixement,pwd) values  
( 'johnQ@themail.cat', 'John', 'Quimet', 'Manresa', '1978-12-25', 'johnQ2578'),  
( 'sarahC@themail.cat', 'Sarah', 'Hidalgo', 'Barcelona', '1984-10-4', 'sarah22'),  
( 'annaL@themail.cat', 'Anna', 'Lleva', 'Manresa', '1990-11-9', 'abracadabra'),  
( 'pereG@mail.com', 'Pere', 'Guzman', 'València', '1970-3-24', '24031970'),  
( 'lluisT@mail.com', 'Lluis', 'Talavera', 'Madrid', '1972-7-8', '08071972');
```

```
insert into amistats (email1,email2,estat) values  
( 'johnQ@themail.cat', 'sarahC@themail.cat', 'Pendent'),  
( 'sarahC@themail.cat', 'pereG@mail.com', 'Pendent'),  
( 'annaL@themail.cat', 'sarahC@themail.cat', 'Pendent'),  
( 'pereG@mail.com', 'johnQ@themail.cat', 'Pendent'),  
( 'lluisT@mail.com', 'johnQ@themail.cat', 'Pendent');
```

# Modificació

instructor(ID,number-ss,name,dept-name,data-inici,data-fi,salary)

**update** instructor

**set** salary = salary \* 1.03

**where** salary > 100000;

**update** instructor

**set** salary = salary \* 1.05

**where** salary <= 100000;

```
mysql> select * from amistats;
```

email1	email2	estat
johnQ@themail.cat	sarahC@themail.cat	Pendent
sarahC@themail.cat	pereG@mail.com	Pendent
annaL@themail.cat	sarahC@themail.cat	Pendent
pereG@mail.com	johnQ@themail.cat	Pendent
lluisT@mail.com	johnQ@themail.cat	Pendent

```
5 rows in set (0.00 sec)
```

**update** usuaris

**set** estat='Aprovada'

**where** email1='annaL@themail.cat' **and**  
email2='sarahC@themail.cat';

## Eliminació de la taula i les tuples

```
drop table student;
```

## Mantenir la taula

```
delete from student;
```

## Eliminar una columna

```
alter table Persons  
drop column DateOfBirth;
```

## Canviar el tipus d'una columna

```
alter table Persons  
alter column DateOfBirth date;
```

## Afegir nova columna

```
alter table Persons  
add DateOfBirth date;
```

## Modificar els valors d'algunes files d'una taula

```
update empleats  
set sou=sou+1000  
where num_proj=44;
```

```
select A1, A2, ..., An  
  from r1, r2, ..., rm  
  where P;
```

- $A_i$  representa un atribut
- $r_i$  representa una relació
- $P$  és un predicat

El resultat és una relació.

instructor(ID,number-ss,name,dept-name,data-inici,data-fi,salary)

**select** name **from** instructor;

Eliminació de duplicats: clàusula distinct

**select distinct** dept\_name **from** instructor;

Manteniment de duplicats: clàusula all

**select all** dept\_name **from** instructor;

# La clàusula select: \* i expressions

```
select * from instructor;
```

Pot contenir expressions +, -, \*, i /, operant amb constants i/o atributs de les tuples.

```
select ID, name, salary/12 from instructor;
```

```
select ID, name, salary/12 as monthly_salary  
from instructor;
```



# La clàusula where i condicions de cerca

Predicat de selecció de l'àlgebra relacional

instructor(ID,number-ss,name,dept-name,data-inici,data-fi,salary)

**select** name

**from** instructor

**where** dept\_name = 'Comp. Sci.' **and** salary > 80000;

- Connector lògics and, or, not
- comparacions aplicades a resultats d'expressions aritmètiques.

```
select * from usuaris where dataNaixement > '1980';
```

```
+-----+-----+-----+-----+-----+-----+
| email          | nom  | cognom | poblacio | dataNaixement | pwd          |
+-----+-----+-----+-----+-----+-----+
| sarahC@themail.cat | Sarah | Hidalgo | Barcelona | 1984-10-04    | sarah22     |
| annaL@themail.cat | Anna  | Lleva   | Manresa  | 1990-11-09    | abracadabra |
+-----+-----+-----+-----+-----+-----+
```

```
2 rows in set, 1 warning (0.00 sec)
```

# Predicats de la clàusula where

instructor(ID,number-ss,name,dept-name,data-inici,data-fi,salary)

teaches(ID, course-id, sec-id, semester, year)

```
select name  
from instructor  
where salary between 90000 and 100000;
```

```
select name, course_id  
from instructor, teaches  
where (instructor.ID, dept_name) = (teaches.ID, 'Biology');
```

# Operacions amb Strings

```
select name  
from instructor  
where name like '%dar%';
```

```
select name  
from instructor  
where name like 'd_____';
```

```
mysql> select nom,cognom,poblacio,dataNaixement from usuaris where nom like 'J%';  
+-----+-----+-----+-----+  
| nom | cognom | poblacio | dataNaixement |  
+-----+-----+-----+-----+  
| John | Quimet | Manresa | 1978-12-25 |  
+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

# Predicat in / not in

```
select nom,ciutat  
from departaments  
where ciutat not in ('Barcelona','Manresa');
```

```
select name  
from instructor  
where salary is null;
```

# Ordenació de tuples

```
select distinct name  
from instructor  
order by name;
```

Default: ascending order

```
order by name desc;
```

```
order by dept_name, name;
```

```
select nom,cognom,poblacio,dataNaixement from usuaris  
where poblacio='Manresa' order by cognom;
```

```
+-----+-----+-----+-----+  
| nom | cognom | poblacio | dataNaixement |  
+-----+-----+-----+-----+  
| Anna | Lleva | Manresa | 1990-11-09 |  
| John | Quimet | Manresa | 1978-12-25 |  
+-----+-----+-----+-----+  
2 rows in set (0.00 sec)
```

section(course-id,sec-id,semester,year,building,room-nbr)

```
(select course_id from section where sem = 'Fall' and year = 2009)  
union  
(select course_id from section where sem = 'Spring' and year = 2010);
```

```
(select course_id from section where sem = 'Fall' and year = 2009)  
intersect  
(select course_id from section where sem = 'Spring' and year = 2010);
```

```
(select course_id from section where sem = 'Fall' and year = 2009)  
except  
(select course_id from section where sem = 'Spring' and year = 2010);
```

- avg: valor mitjà d'una columna
- min: valor mínim d'una columna
- max: valor màxim d'una columna
- sum: suma els valors d'una columna
- count: número total de files seleccionades



instructor(ID,number-ss,name,dept-name,data-inici,data-fi,salary)

teaches(ID, course-id, sec-id, semester, year)

```
select avg (salary)
from instructor
where dept_name= 'Comp. Sci.';
```

```
select count (distinct ID)
from teaches
where semester = 'Spring' and year = 2010;
```

```
select count (*)
from course;
```

# Funcions d'agregació: group by

instructor(ID,number-ss,name,dept-name,data-inici,data-fi,salary)

```
select dept_name, avg (salary)
from instructor
group by dept_name;
```

Els departaments sense instructor no apareixeran en el resultat

```
/* Consulta erronia */
select dept_name, ID, avg (salary)
from instructor
group by dept_name;
```

# Funcions d'agregació: la clàusula Having

instructor(ID,number-ss,name,dept-name,data-inici,data-fi,salary)

Cerca dels noms i salari promig de tots els departaments amb un salari promig superior a 42000

```
select dept_name, avg (salary)
from instructor
group by dept_name
having avg (salary) > 42000;
```

Els predicats en la clàusula having s'apliquen després de la formació de grups, mentre que els predicats de la clàusula where s'apliquen abans de formar els grups.

```
select sum (salary )  
from instructor;
```

- Totes les operacions d'agregació excepte count(\*) ignoren tuples amb valor null values en els atributs d'agregació
- Què passa si una col·lecció de valors només conté valors null?
  - count retorna 0
  - la resta d'agregacions retorna null

- Ús de subconsultes per testeig de pertinença, comparació de conjunts i cardinalitat de conjunts

# Subconsultes niuades: Exemples

section(course-id, sec-id, semester, year, building,  
room-number, time-slot-id)

```
select distinct course_id
from section
where semester = 'Fall' and year= 2009 and
      course_id not in (select course_id
                        from section
                        where semester = 'Spring'
                        and year= 2010);
```

# Subconsultes niuades: Exemples

takes(ID, course-id, sec-id, semester, year, grade)  
teaches(ID, course id, sec id, semester, year)

```
select count (distinct ID)  
from takes  
where (course_id, sec_id, semester, year) in  
      (select course_id, sec_id, semester, year  
      from teaches  
      where teaches.ID= 10101);
```

# Subconsultes niuades: Exemples

instructor(ID, name, dept name, salary)

```
select distinct T.name  
from instructor as T, instructor as S  
where T.salary > S.salary and S.dept_name = 'Biology';
```

```
select name  
from instructor  
where salary > some (select salary  
                    from instructor  
                    where dept_name = 'Biology');
```



instructor(ID, name, dept name, salary)

```
select name
from instructor
where salary > all (select salary
                    from instructor
                    where dept_name = 'Biology');
```

section(course id, sec id, semester, year, building,  
room-number, time-slot-id)

```
select course_id
from section as S
where semester = 'Fall' and year= 2009 and
exists (select *
from section as T
where semester = 'Spring' and year= 2010
and S.course_id= T.course_id);
```

# Not exists

student(ID, name, dept-name, tot-cred)  
course(course-id, title, dept-name, credits)  
takes(ID, course-id, sec-id, semester, year, grade)

```
select distinct S.ID, S.name  
from student as S  
where not exists ( (select course_id  
                    from course  
                    where dept_name = 'Biology')  
except  
                  (select T.course_id  
                   from takes as T  
                   where S.ID = T.ID));
```

section(course-id, sec-id, semester, year, building,  
room-number, time-slot-id)  
course(course-id, title, dept-name, credits)

```
select T.course_id  
from course as T  
where unique (select R.course_id  
                from section as R  
                where T.course_id= R.course_id  
                    and R.year = 2009);
```

# Subconsultes en la clàusula from

instructor(ID, name, dept name, salary)

```
select dept_name, avg_salary
from (select dept_name, avg (salary) as avg_salary
      from instructor
      group by dept_name)
where avg_salary > 42000;
```

```
select dept_name, avg_salary
from (select dept_name, avg (salary)
      from instructor
      group by dept_name)
      as dept_avg (dept_name, avg_salary)
where avg_salary > 42000;
```

instructor(ID, name, dept name, salary)  
department(dept-name, building, budget)

```
select dept_name,  
        (select count(*)  
        from instructor  
        where department.dept_name = instructor.dept_name)  
as num_instructors  
from department;
```

```
select name  
        from instructor  
        where salary * 10 >  
        (select budget from department  
        where department.dept_name = instructor.dept_name);
```

Runtime error if subquery returns more than one result tuple

Eliminació de tots

```
delete from instructor ;
```

instructor(ID, name, dept name, salary)

department(dept-name, building, budget)

Eliminació dels d'un departament determinat

```
delete from instructor  
where dept_name= 'Finance';
```

```
delete from instructor where dept_name in  
  (select dept_name  
   from department  
   where building = 'Watson');
```

```
delete from instructor  
where salary < (select avg (salary) from instructor);
```

Problemes: A mesura que eliminem tuples, el promig canvia

Solució: 1. Primer, càlcul avg salary i cerca de les tuples a eliminar  
2. Eliminar les tuples trobades en el pas anterior (sense recalcular avg o retesteant les tuples)



```
insert into course values ('CS-437', 'Database Systems', 'Comp. Sci.', 4);
```

```
insert into course (course_id, title, dept_name, credits)  
  values ('CS-437', 'Database Systems', 'Comp. Sci.', 4);
```

```
insert into student values ('3003', 'Green', 'Finance', null);
```

```
insert into student  
select ID, name, dept_name, 0  
from instructor;
```

La sentència `select from where` statement s'avalua completament abans que qualsevol dels resultats s'insereixin en la relació (altrament provocaria problemes si per exemple `table1` no tingués clau primària definida)

```
insert into table1 select * from table1;
```

```
update instructor  
  set salary = salary * 1.03  
  where salary > 100000;  
update instructor  
  set salary = salary * 1.05  
  where salary <= 100000;
```

L'ordre és important

```
update instructor set salary = case  
  when salary <= 100000 then salary * 1.05  
  else salary * 1.03  
end;
```

- Capítol IV, Sistac et al, “Bases de Dades”.
- Benavides et al, “SQL Para usuarios y programadores”, Ed. Paraninfo.
- Capítol III, Silberschatz, Korth and Sudarshan, “Database system Concepts. 6th edition” (pdf available)