Bases de Datos Relacionales

Introducción a SQL

Introducción a SQL

Introducción a SQL

- DDL
  - Especifica la estructura de una base de datos
  - Define
    - Esquema de cada tabla o relación
    - Los tipos de valores de cada atributo
    - Las restricciones de integridad
    - Los índices de cada relación
    - La seguridad para el acceso
    - El almacenamiento físico

Introducción a SQL

- DDL
  - Tipos de datos comunes
    - char(n)
    - varchar(n)
    - int
    - numeric(p,d)
    - real
    - float(n)
    - date
    - time

Introducción a SQL

- DDL
  - CREATE

create table r
$(A_1 \quad D_1,$
$A_2 \quad D_2,$
$\ldots,$
$A_n \quad D_n,$
$\langle$integrity-constraint$_1\rangle,$
$\ldots,$
$\langle$integrity-constraint$_k\rangle);$

create table *department*
(*dept_name* **varchar** (20),
building **varchar** (15),
budget **numeric** (12,2),
primary key (*dept_name*));

Introducción a SQL

- DDL
  - CREATE
    - Llave primaria
      - Identifican de manera única un registro de la tabla
      - Debe ser Not null y única
    - Llave foránea
      - Identifican de manera única un registro de otra tabla
      - Debe ser Not null

Introducción a SQL

- DDL
  - CREATE

```
create table department
    (dept_name      varchar (20),
     building       varchar (15),
     budget         numeric (12,2),
     primary key (dept_name));

create table course
    (course_id      varchar (7),
     title          varchar (50),
     dept_name      varchar (20),
     credits        numeric (2,0),
     primary key (course_id),
     foreign key (dept_name) references department);
```

```
create table instructor
    (ID             varchar (5),
     name           varchar (20) not null,
     dept_name      varchar (20),
     salary         numeric (8,2),
     primary key (ID),
     foreign key (dept_name) references department);

create table section
    (course_id      varchar (8),
     sec_id         varchar (8),
     semester       varchar (6),
     year           numeric (4,0),
     building       varchar (15),
     room_number    varchar (7),
     time_slot_id   varchar (4),
     primary key (course_id, sec_id, semester, year),
     foreign key (course_id) references course);
```

Introducción a SQL

- DDL
  - CREATE

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

Introducción a SQL

- DDL
  - DROP
  - ALTER

drop table r;

alter table r **add** A D;

alter table r **drop** A;

Introducción a SQL

- DML
  - INSERT

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

Introducción a SQL

- DML
  - UPDATE

```
update instructor
set salary= salary * 1.05;

update instructor
set salary = salary * 1.05
where salary < 70000;
```

Introducción a SQL

- DML
  - DELETE

```
delete from instructor
where salary between 13000 and 15000;

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

Introducción a SQL

- DML
  - SELECT

```
select name
from instructor;
```

```
select distinct dept_name
from instructor;
```

```
select ID, name, dept_name, salary * 1.1
from instructor;
```

```
select name
from instructor
where dept_name = 'Comp. Sci.' and salary > 70000;
```

Introducción a SQL

- DML
  - SELECT

```
select name, instructor.dept_name, building
from instructor, department
where instructor.dept_name= department.dept_name;
```

```
select instructor.*
from instructor, teaches
where instructor.ID= teaches.ID;
```

Introducción a SQL

- DML
  - Natural Join

```
select name, course_id
from instructor, teaches
where instructor.ID= teaches.ID;

select name, course_id
from instructor natural join teaches;
```

Introducción a SQL

- DML
  - Rename

```
select T.name, S.course_id
from instructor as T, teaches as S
where T.ID= S.ID;
```

Introducción a SQL

- DML
  - Like

```
select dept_name
from department
where building like '%Watson%';
```

Introducción a SQL

- DML
  - Order

```
select name
from instructor
where dept_name = 'Physics'
order by name;
```

```
select *
from instructor
order by salary desc, name asc;
```

Introducción a SQL

- DML
  - Where predicates

```
select name
from instructor
where salary <= 100000 and salary >= 90000;
```

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

Introducción a SQL

- DML
  - Set Operations: Union

```
select course_id
from section
where semester = 'Fall' and year= 2009;
```

```
select course_id
from section
where semester = 'Spring' and year= 2010;
```

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

Introducción a SQL

- DML
  - Set Operations: Intersect

```
select course_id
from section
where semester = 'Fall' and year= 2009;
```

```
select course_id
from section
where semester = 'Spring' and year= 2010;
```

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

Introducción a SQL

- DML
  - Set Operations: Except

```
select course_id
from section
where semester = 'Fall' and year= 2009;
```

```
select course_id
from section
where semester = 'Spring' and year= 2010;
```

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

Introducción a SQL

- DML
  - Aggregate Functions (avg, min, max, sum, count)

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

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

Introducción a SQL

- DML
  - Group by

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

Introducción a SQL

- DML
  - Having

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

Introducción a SQL

- DML
  - Nested sub-queries

```
select distinct course_id
from section
where semester = 'Fall' and year= 2009 and
        course_id in (select course_id
                      from section
                      where semester = 'Spring' and year= 2010);
```

Introducción a SQL

- DML
  - Nested sub-queries

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

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

Introducción a SQL

- DML
  - Nested sub-queries

```
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);
```

Introducción a SQL

- DML
  - Nested sub-queries

```
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));
```

Introducción a SQL

- DML
  - Nested sub-queries

```
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);
```

Introducción a SQL

- DML
  - Nested sub-queries

```
select T.course_id
from course as T
where not unique (select R.course_id
                  from section as R
                  where T.course_id= R.course_id and
                        R.year = 2009);
```

Introducción a SQL

- DML
  - Nested sub-queries

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

Introducción a SQL

- DML
  - Nested sub-queries

```
with max_budget (value) as
        (select max(budget)
        from department)
select budget
from department, max_budget
where department.budget = max_budget.value;
```

Bases de Datos Relacionales

SQL Intermedio