Basic SQL Commands

Select 'Hello, world';

Select 1+2;

Show tables;

Describe [table_name];

- Shows the field details

Select * from [table_name];

Select count(*) from [table_name];

- Shows the number of rows

Select * from [table_name] Limit 5;

- returns 5 rows

Select * from [table_name] LIMIT 5 OFFSET 5;

- returns next 5 rows

show table status;

- Shows the information of all tables

show table status like 'test';

show create table test;

Select * from test where a = NULL;

- will not work because null is not a value

Select * from test where a is NULL;

Select * from test where a is not NULL;

SELECT with WHERE condition

Import ‘university’ Database:

```
select dept_name from instructor;
```

```
select all dept_name from instructor;
```

- 'all' is default

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

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

- It does not result in any change to the instructor relation

Import ‘world’ Database:

```
Select name, population / 1000000 as PopMM
        From country
        Where population >= 1000000
        Order by population desc;
```

```
SELECT Name, Continent, Population
        FROM Country
        WHERE Name LIKE '%island%' ORDER BY Name;
```

```
SELECT Name, Continent, Population
        FROM Country
        WHERE Continent IN ( 'Europe', 'Asia' );
```

Create a Table

```
CREATE TABLE test (a int, b text, c text);

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 (dept_name));
```

Insert Data in a Table

```
INSERT INTO test VALUES (2, 'Test', 'Second'), (3, 'Third', 'Row');
INSERT INTO test (b, c) VALUES ('what', 'when');
INSERT INTO test (a, b, c) VALUES (0, null, '');
INSERT INTO test (a, b, c) SELECT id, name, description FROM item;
```

UPDATE a Table

```
UPDATE test SET c='hello' WHERE a=2;
```

Alter Table

```
ALTER TABLE r ADD a d;
```

- To add attributes to an existing relation
- All tuples in the relation are assigned null as the value for the new attribute.
- r is the name of an existing relation, a is the name of the attribute to be added, and d is the type of the added attribute

```
ALTER TABLE r DROP a;
```

- Deletes column a

```
ALTER TABLE t1 RENAME t2;
ALTER TABLE t2 MODIFY a TINYINT NOT NULL, CHANGE b c CHAR(20);
```

- To change column a from INTEGER to TINYINT NOT NULL (leaving the name the same), and to change column b from CHAR(10) to CHAR(20) as well as renaming it from b to c

```
ALTER TABLE ‘Student’
        ADD PRIMARY KEY(‘id’);
```

Delete and Drop

```
DELETE FROM r;
```

- Keeps relation r
- but deletes all tuples in r.

```
DELETE FROM test WHERE a=2;
```

```
DROP TABLE r;
```

- Deletes not only all tuples of r, but also the schema for r.
- After r is dropped, no tuples can be inserted into r unless it is re-created with the create table command.

```
DROP TABLE IF EXISTS [table_name];
```

```
DROP DATABASE [database_name];
```

Cartesian Product

```
select * from a, b

select * from instructor, department
```

Queries on Multiple Relations

Task:

Retrieve the names of all instructors, along with their department names and department building name.

Schema:

instructor(ID, name, dept_name, salary)
department(dept_name, building, budget)

SQL:

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