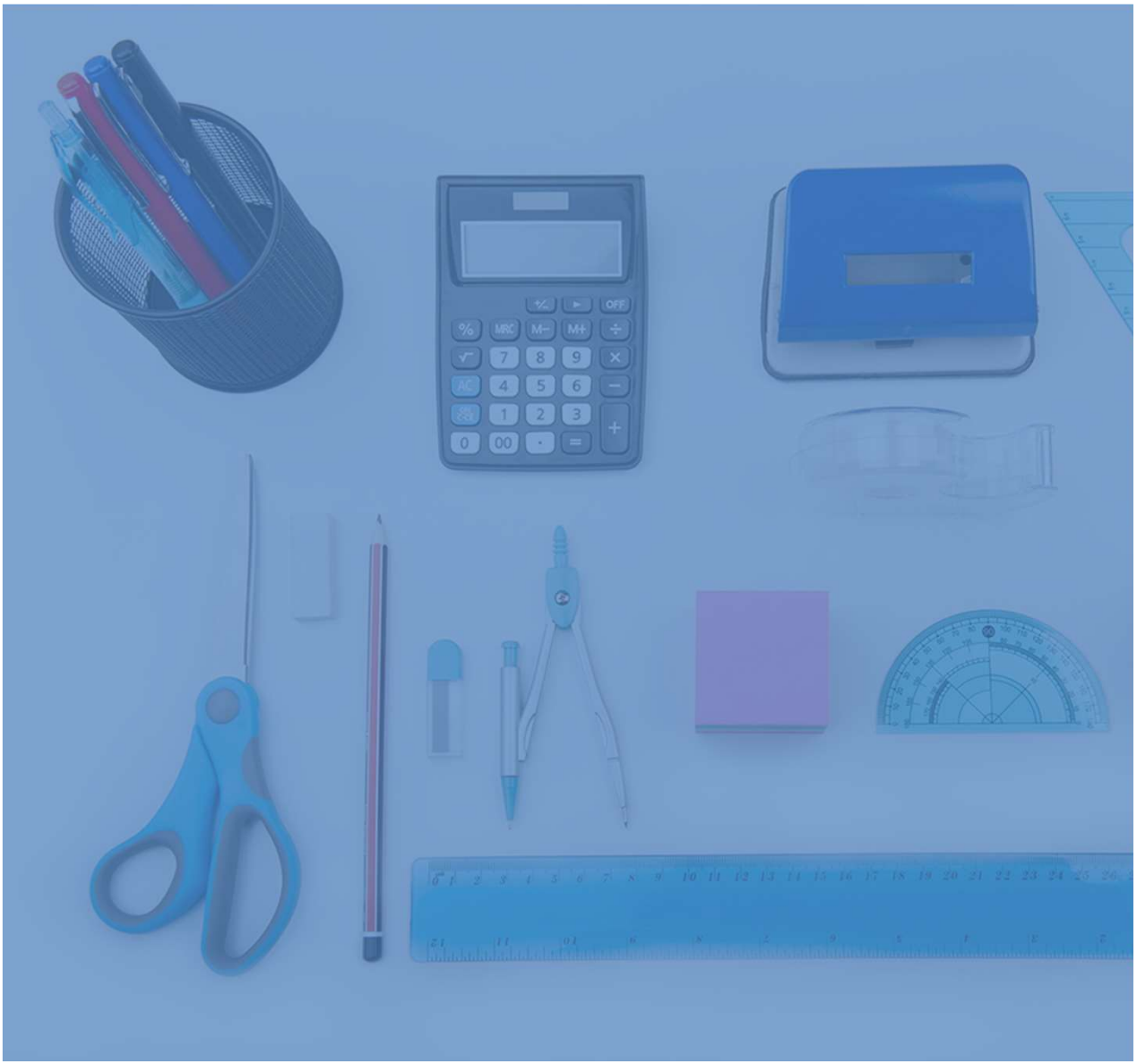
Week-5

Introduction to Database Systems

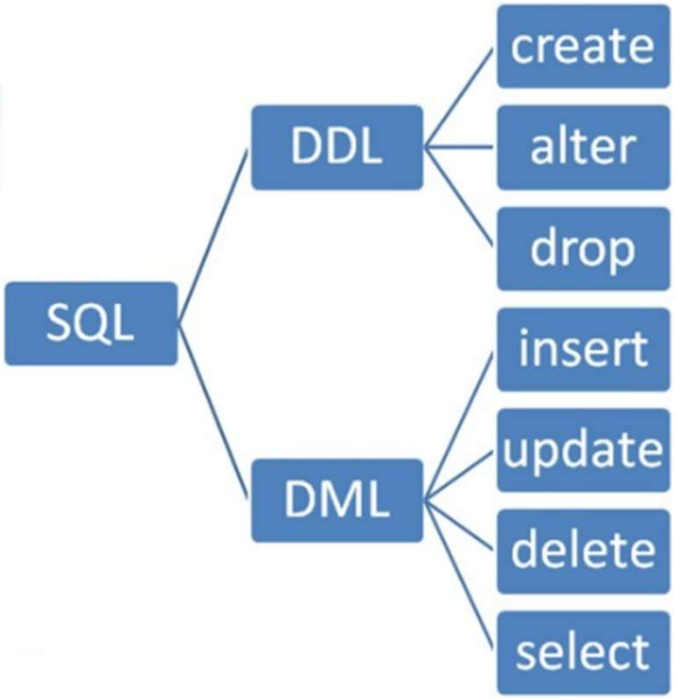
Shah Nawaz



Introduction to SQL (DDL)

Basic SQL (Tool: MYSQL Workbench)

SQL
Data Definition Langage
Data Manipulation Language



Main Commands for DDL

- Using DDL commands we can make changes in the structure of the database.

- Main Commands:
  - Create – used to create a new table in the database
  - Alter – used to make changes in the already created tables
  - Drop – used to delete an existing table

Create Table

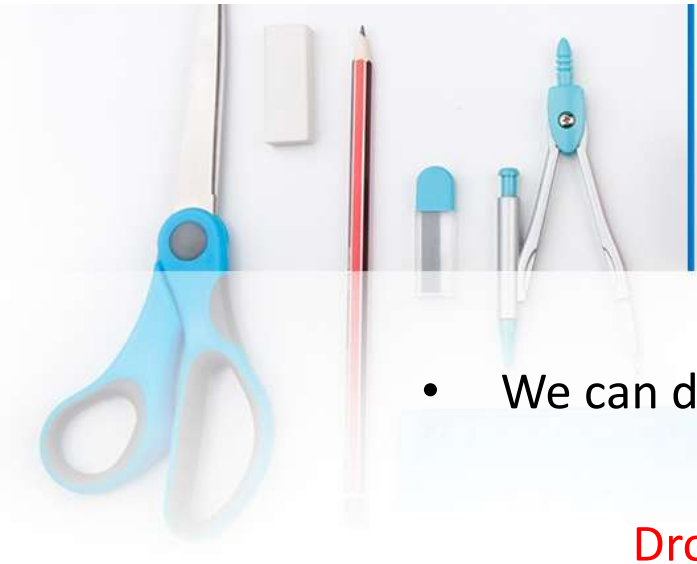
- Before creating a table in the database, we need to create a database itself:

```
Create database databaseName;
```

Example:

```
Create database HealthCareSystem;
```

- SQL is not case sensitive.
- Every command needs to end with a semicolon.



Delete Table

- We can delete a database :

```
Drop database DatabaseName;
```

Example:

```
Drop database HealthCareSystem;
```

Getting Started …

- Before creating a new table or viewing already constructed tables we need to make sure we are in scope of the database or we are present in a database.

  Use databaseName ;

- Example:

  Use HealthCareSystem;

- We can see tables present in an existing database:
- Example:

  Show tables;

How to create a table Datatypes & Constraints

Datatypes

Numeric values

- INT (Size: 4 Bytes)
- NUMERIC(I, j) 888.88
- Fixed point number, with user-specified precision of *p* digits, with *d* digits to the right of decimal point

Character String

- Fixed length: CHAR(5) CAT_ _
- Non-fixed length: VARCHAR(n) CAT

Some more important datatypes

- Boolean: TRUE, FALSE, NULL
- DATE : yyyy-mm-dd
- TIME: hh:mm:ss
- TIMESTAMP: DATE(DD-MM-YYYY) & TIME (HH:MM:SS)

Key words for Constraints

- **NOT NULL** – used for required fields
- **UNIQUE** – used for attributes that cannot have a duplicate value (other than pk)
- **DEFAULT** – used to assign some default value to an attribute
- **CHECK(Dnumber > 0 AND Dnumber < 21)** -- used to set a condition on the input value for an attribute. E. g age
- **Primary key** – used to make an attribute primary key of the table
- **Foreign Key** *(A)* **references** *s(B)* -- used to make an attribute foreign key.

Create Table

ENTITY = Employee

ATTRIBUTES

- **First Name,**
- **Middle Name,**
- **Last Name,**
- **Social Security Number,**
- **Date of Birth,**
- **Address,**
- **Gender,**
- **Salary,**
- **Social Security Number ,**
- **Department number**

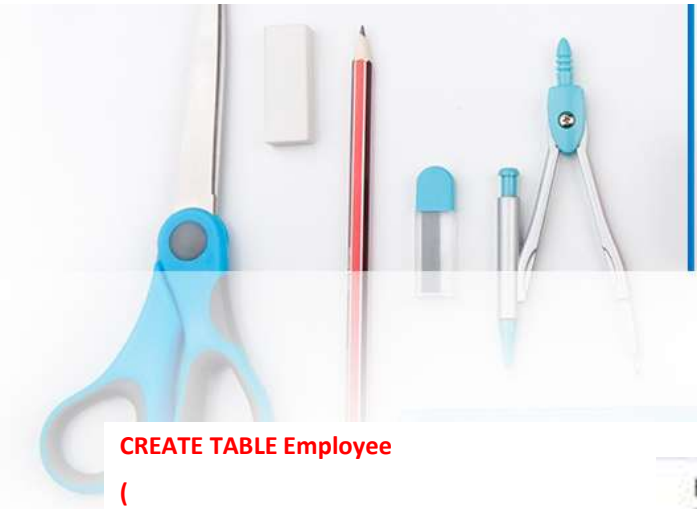
Create Table

```
CREATE TABLE tablename
(
  AttributeName Datatype Constraint,
                ....
                ...
);
```

Note: Constraint is optional. It can be written here or may be later as well

Employee

```
CREATE TABLE Employee
(
        Fname VARCHAR(15) NOT NULL,
        Minit CHAR(1),
        Lname VARCHAR(15) NOT NULL,
        Ssn CHAR (9) NOT NULL,
        Bdat DATE,
        Address VARCHAR(30)  DEFAULT 'UCP',
        Gender Char(1),
        Salary int,
        Super_ssn CHAR(9),
        Dno INT
);
```



Employee

```
CREATE TABLE Employee
(
        Fname VARCHAR(15) NOT NULL,
        Minit CHAR(1),
        Lname VARCHAR(15) NOT NULL,
        Ssn CHAR (9) NOT NULL,
        Bdat DATE,
        Address VARCHAR(30)  DEFAULT 'UCP',
        Gender Char(1),
        Salary int,
        Super_ssn CHAR(9),
        Dno INT
);
```

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	gen	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

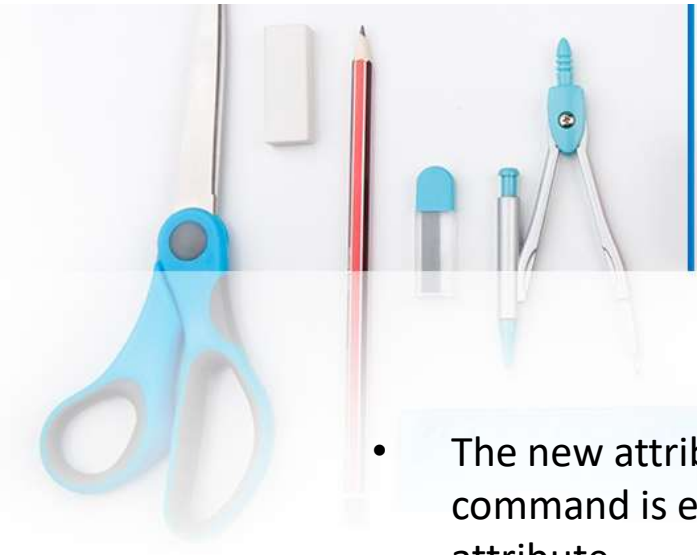
DROP TABLE

- Used to remove a relation (table) *and its definition*
- The relation can no longer be used in queries, updates, or any other commands since its description no longer exists
- Example:

```
DROP TABLE  Employee;
```

ALTER TABLE

- Used to add an attribute to one of the tables
- Used to drop an existing attribute of the table
- Used to change the datatype of an existing attribute of the table
- Used to change the name of an existing attribute of the table
- Used to add/drop constraints



Add/Drop attribute

- The new attribute will have NULLs in all the tuples of the relation right after the command is executed; hence, the NOT NULL constraint is *not allowed* for such an attribute
- Example:

```
ALTER TABLE EMPLOYEE ADD JOB VARCHAR(12);
```

-

```
ALTER TABLE EMPLOYEE DROP JOB ;
```

Setting position for a new attribute

- By default attribute is added at the end of the table

student

Sid	Name	Address

ALTER TABLE student ADD Email varchar(50) FIRST;

ALTER TABLE student DROP Email;

ALTER TABLE student ADD Email varchar(50) AFTER Sid;

Changing a Column Datatype or Name

```
ALTER TABLE student Modify Email varchar(100);
```

```
ALTER TABLE student Change Email Email_Address varchar(100);
```

Change command can change both datatype and name.

```
ALTER TABLE student Change Email Email_Address varchar(50);
```

The main difference between `modify` and `change` in SQL is that `change` can rename a column, while `modify` cannot:

Adding Not Null and Default Value of Attributes

ALTER TABLE student MODIFY Address NOT NULL;

ALTER TABLE student MODIFY Address DEFAULT 'LAHORE';

- TO change Default value:

  ALTER TABLE student ALTER Address SET DEFAULT 'KARACHI';

- You can remove default constraint from any column:

  ALTER TABLE student ALTER Address drop default;

Rename a table

Existing table name can be changed as:

```
ALTER TABLE Student RENAME TO UnderGrad_student;
```

Add / Drop primary key

Add Primary Key:

Alter table employee Add primary key (ssn);

Drop Primary Key:

Alter table employee Drop primary key;

Add / Drop primary key

Example 1

```
CREATE TABLE students (
    student_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    birth_date DATE
);
```

Example 2

```
CREATE TABLE orders (
    order_id INT,
    product_id INT,
    quantity INT,
    PRIMARY KEY (order_id, product_id)
);
```

Add / Drop foreign key

```
CREATE TABLE departments (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(50)
);

CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    employee_name VARCHAR(100),
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES departments(department_id)
);
```

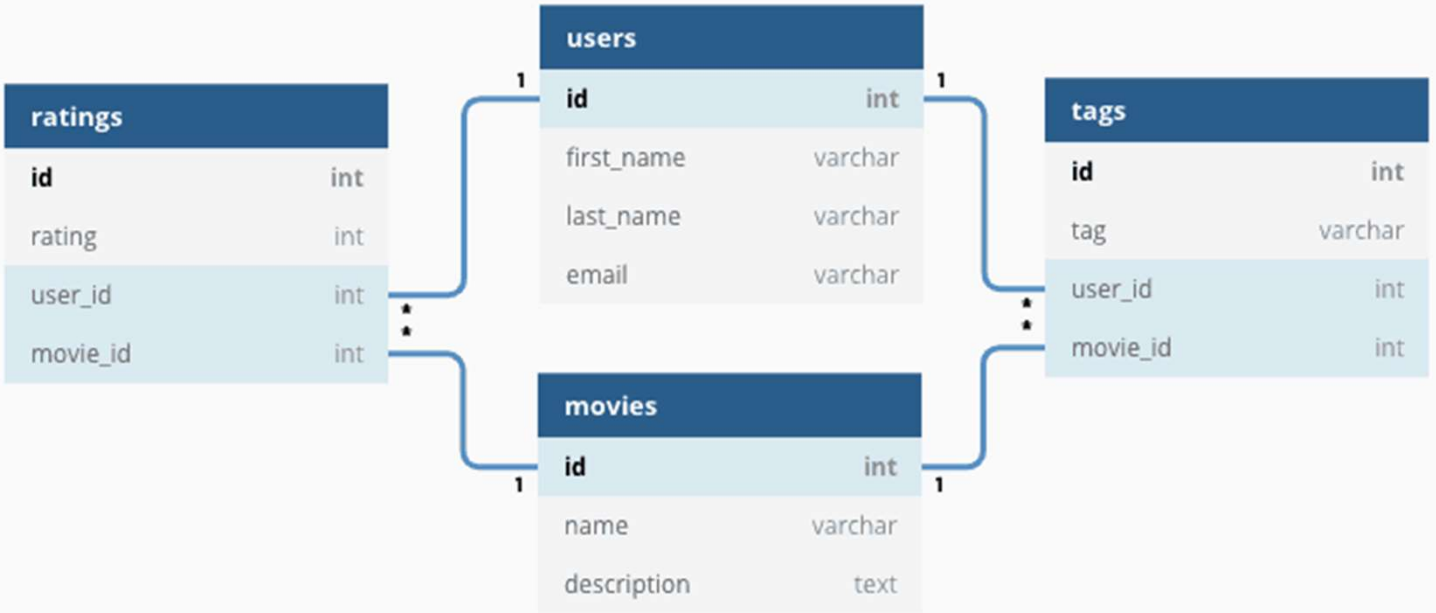
Add / Drop foreign key

Alter table employee Add constraint fk_1 foreign key (dno) references department (dnumber);

Alter table employee Drop foreign key fk_1;

Class Activity

Create tables



Data Manipulation Language (DML)

Commands: Insert, Delete, Update & select

Specifying Updates in SQL

- There are three SQL commands to modify the database;
  - **INSERT,**
  - **DELETE, and**
  - **UPDATE**

Adding values in table

```
-- Creating a 'departments' table ----------
CREATE TABLE departments (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(50)
);
```

Attribute values should be listed in the same order as the attributes were specified in the CREATE TABLE command

```
-- Inserting sample data ----------
INSERT INTO departments (department_id, department_name)
VALUES    (1, 'HR'),   (2, 'Finance');
```

INSERT

- Insert a tuple for a new EMPLOYEE for whom we only know the FNAME, LNAME, and SSN attributes.

INSERT INTO EMPLOYEE (FNAME, LNAME, SSN)
VALUES ('Richard', 'Marini', 653298653), (‘Ali’, ‘Ahmed’ , 123456889);

DELETE with condition [WHERE]

- Removes tuples from a relation
- Includes a WHERE-clause to select the tuples/rows to be deleted

Examples:

```
1.  DELETE FROM  EMPLOYEE
        WHERE    LNAME='Brown';
2.  DELETE FROM  EMPLOYEE
        WHERE    SSN='123456789';
3.  DELETE FROM  EMPLOYEE
        WHERE    DNO  =5;
```

- The number of tuples deleted depends on the number of tuples in the relation that satisfy the WHERE-clause

Delete all records of table

- A missing WHERE-clause specifies that *all tuples* in the relation are to be deleted; the table then becomes an empty table

DELETE FROM EMPLOYEE;

Delete [Specific record]

- Tuples are deleted from only *one table* at a time (unless CASCADE is specified on a referential integrity constraint)

```
Delete from employee
Where ssn = 123456789;
```

UPDATE [one or more than one values]

- Used to modify attribute values of one or more selected tuples
- A WHERE-clause selects the tuples to be modified
- An SET-clause specifies the attributes to be modified and their new values

- Example: Change the FNAME to AAMIR and Age to 19 for a student whose RollNo is 10.

- **UPDATE STUDENT**
  SET FNAM = 'AAMIR', AGE = 19
  WHERE ROLLNO=10;

UPDATE [Modify values with formula or calculation]

- Example: Give all employees in the 'Research' department a 10% raise in salary.

```
UPDATE EMPLOYEE
        SET SALARY = SALARY *1.1
        WHERE Dno = 5;
```

- In this request, the modified SALARY value depends on the original SALARY value in each tuple

Retrieval Data from Table (s): SELECT statement

- SQL has one basic statement for retrieving information from a database

SELECT	< attribute list >	
FROM	< table (s) >	
WHERE	< condition >	(Optional)

  - <attribute list> is a list of attribute names whose values are to be retrieved by the query
  - <table list> is a list of the relation names required to process the query
  - <condition> is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query

Simple SQL Queries (cont.)

- **Query:** **Retrieve the name and address of all employees.**

```
SELECT    FNAME, MINIT, LNAME, ADDRESS
FROM      EMPLOYEE ;
```

FNAME	MINIT	LNAME	ADDRESS
John	B	Smith	731 Fondren, Houston, TX
Franklin	T	Wong	638 Voss, Houston, TX
Alicia	J	Zelaya	3321 Castle, Spring, TX
Jennifer	S	Wallace	291 Berry, Bellaire, TX
Ramesh	K	Narayan	975 Fire Oak, Humble, TX
Joyce	A	English	5631 Rice, Houston, TX
Ahmad	V	Jabbar	980 Dallas, Houston, TX
James	E	Borg	450 Stone, Houston, TX

Simple SQL Queries (cont.)

- Query: Retrieve the information of all employees.

```
SELECT   *
FROM     EMPLOYEE ;
```

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

Simple SQL Queries

•Query : Retrieve the birthdate and address of the employee whose name is 'John B. Smith'.

```
SELECT   BDATE, ADDRESS
         FROM      EMPLOYEE
         WHERE     FNAME='John'
         AND       MINIT='B'
         AND       LNAME='Smith';
```

BDATE	ADDRESS
1965-01-09	731 Fondren, Houston, TX

Insert / Cascade Update

```
-- Creating a 'departments' table
CREATE TABLE departments (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(50)
);
-- Creating an 'employees' table with a foreign key and ON UPDATE CASCADE
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    employee_name VARCHAR(100),
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES departments(department_id)
        ON UPDATE CASCADE
);
```

```
-- Inserting sample data
INSERT INTO departments
(department_id, department_name) VALUES
    (1, 'HR'),
    (2, 'Finance');


INSERT INTO employees
(employee_id, employee_name, department_id)
VALUES
    (101, 'John Doe', 1),
    (102, 'Jane Smith', 2);
```

Delete / Cascade Delete

```
-- Creating a 'departments' table
CREATE TABLE departments (
    department_id INT PRIMARY KEY,
    department_name VARCHAR(50)
);

-- Creating an 'employees' table with a foreign key and ON UPDATE CASCADE
CREATE TABLE employees (
    employee_id INT PRIMARY KEY,
    employee_name VARCHAR(100),
    department_id INT,
    FOREIGN KEY (department_id) REFERENCES departments(department_id)
        ON UPDATE CASCADE
);
```

```
-- Inserting sample data
INSERT INTO departments (department_id, department_name) VALUES
    (1, 'HR'),
    (2, 'Finance');

INSERT INTO employees (employee_id, employee_name, department_id) VALUES
    (101, 'John Doe', 1),
    (102, 'Jane Smith', 2);
```

```
-- Deleting a department and its associated employees
DELETE FROM departments
WHERE department_id = 1;
```

Thank You all!