SQL: Structured Query Language

DDL(Data Definition Language):used to modify the structure of the table (schema)

- CREATE TABLE
- DROP TABLE
- ALTER TABLE

DML(Data Manipulation Language):used to modify the records of the table

- INSERT INTO
- DELETE FROM
- UPDATE SET

DQL(Data Query Language):used to retrieve data from the database

- SELECT
- FROM WHERE
- GROUP BY
- HAVING

DCL(Data Control Language): used transaction intensive data operations

- COMMIT
- ROLLBACK
- TCC

Definitions

- **Candidate key:** Min set of attributes used to differentiate records of the relation. Can be set to NULL.
- **Simple candidate key**
- **Compound candidate key**
- **Prime attribute**
- **Non-prime attribute**
- **Primary key:** Candidate key that has no NULL values.
- **Alternate key/Secondary key:** All candidate keys except primary key. NULL values allowed

- **NULL:**
  - NULL is not 0
  - No two NULL values are equal
  - It is sequence of random ASCII char assigned by DBMS. Not known to user.
- Foreign key:
  - Attribute set that are used to establish a link between data in two tables.
  - The columns used to create the primary key in one table are also used to create the foreign key constraint and can be used to reference data in the same table or in another table.
  - A foreign key can reference either the same table or in another table.
  - Null values are allowed
  - Need not be unique

Student Table: (referenced rel)

sid	user_name
23	Tom
25	Harry
31	Rhim

Course Table: (referencing rel)

sid	cid
23	c1
23	c2
31	c2

Referential Integrity (on referenced rel)

OPERATIONS

- Insertion: (no violation)
- Deletion: (violation)
- Updation:(violation)

ACTIONS

- CASCADE
- SET NULL
- SET DEFAULT

```
CREATE TABLE DEPT (
        DNAME           VARCHAR(10)     NOT NULL,
        DNUMBER         INTEGER         NOT NULL,
        MGRSSN          CHAR(9),
        MGRSTARTDATE    CHAR(9),
        PRIMARY KEY (DNUMBER),
        UNIQUE (DNAME),
        FOREIGN KEY (MGRSSN) REFERENCES EMP
       ON DELETE SET DEFAULT ON UPDATE CASCADE);
```

Aliases

- In SQL, we can use the same name for two (or more) attributes as long as the attributes are in *different relations*
  - **Eg. EMPLOYEE.**LNAME, **DEPARTMENT.**DNAME
- Aliasing can also be used in any SQL query for convenience
- Can also use the AS keyword to specify aliases
- **SELECT** E.FNAME, E.LNAME, S.FNAME, S.LNAME

  FROM EMPLOYEE AS E, EMPLOYEE AS S

  WHERE E.SUPERSSN=S.SSN

SET operations

- UNION / UNION ALL
- INTERSECT / INTERSECT ALL
- MINUS / MINUS ALL

Nested Query

Independent Nested Query	**Co-related Nested Query**
Inner query independent of the outer query	Inner query uses attributes defined in the outer query
bottom - top	top-bottom
IN/NOT IN ANY ALL	EXIST/ NOT EXIST

Insert

Eg: INSERT INTO EMPLOYEE

VALUES ('Richard','K','Marini', '653298653', '30-DEC-52', '98 Oak,Forest,Katy,TX', 'M', 37000,'987654321', 4)

- **An alternate form of INSERT specifies explicitly the attribute names that correspond to the values in the new tuple**

Eg: INSERT INTO EMPLOYEE (FNAME, LNAME,SSN)

VALUES ('Richard', 'Marini', '653298653')

Delete

U4A: DELETE FROM EMPLOYEE

WHERE LNAME='Brown’

U4B: DELETE FROM EMPLOYEE

WHERE SSN='123456789’

update

Eg : UPDATE PROJECT

SET PLOCATION = 'Bellaire', DNUM = 5

WHERE PNUMBER=10