SQL Outline

- Data definition
- Query
  - Basic queries
  - More complex queries
  - Aggregate functions
  - Examples and Practice
- Data update
- View definition
A SQL query consists of SELECT, FROM and a set of optional clauses

```
SELECT <attribute or function list>
FROM <table list>
[WHERE <selection or join condition>]
[GROUP BY <grouping attributes>]
[HAVING <group condition>]
[ORDER BY <attribute list>]
```
Practice

- P5. Find the names of departments with 2 or more male employees
- P6. Find the names of employees who have more than 2 dependents and work on more than 2 projects (hint: intersection)
- P7. Find the names of employees with the most number of dependents
- P8. Find names of employees who work on all projects ‘John Smith’ is working on
Outline

- Data definition
- Working with SQL*Plus
- Query
- Data update
- View definition
Specifying Updates in SQL

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

- It is used to add one or more tuples to a relation
- Attribute values should be listed in the same order as the attributes were specified in the CREATE TABLE command
- **Example:**

  ```
  INSERT INTO EMPLOYEE
  VALUES ('Richard','K','Marini', '653298653', '30-DEC-52', '98 Oak Forest,Katy,TX', 'M', 37000,'987654321', 4 )
  ```
INSERT (cont.)

- Can specify explicitly the attribute names that correspond to the values in the new tuple

- **Example:** 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')
  ```

- The constraints specified in the DDL commands are automatically enforced by the DBMS when updates are applied to the database
Another variation of INSERT allows insertion of *multiple tuples* resulting from a query into a relation.

**Example:** Suppose we want to create a temporary table that has the name, number of employees, and total salaries for each department. A table `DEPTS_INFO` is created by U3A, and is loaded with the summary information retrieved from the database by the query in U3B.

**U3A:**
```
CREATE TABLE DEPTS_INFO
(DEPT_NAME VARCHAR(10),
 NO_OF_EMPS INTEGER,
 TOTAL_SAL INTEGER);
```

**U3B:**
```
INSERT INTO DEPTS_INFO (DEPT_NAME,
 NO_OF_EMPS, TOTAL_SAL)
SELECT DNAME, COUNT (*), SUM (SALARY) FROM DEPARTMENT, EMPLOYEE WHERE DNUMBER=DNO GROUP BY DNAME ;
```

**Note:** The `DEPTS_INFO` table may not be up-to-date if we change the tuples in either the `DEPARTMENT` or the `EMPLOYEE` relations *after* issuing U3B. We have to create a view (see later) to keep such a table up to date.
DELETE

- Removes tuples from a relation
- Includes a WHERE-clause to select the tuples to be deleted
- Tuples are deleted from only one table at a time (unless CASCADE is specified on a referential integrity constraint)
- A missing WHERE-clause specifies that all tuples in the relation are to be deleted; the table then becomes an empty table
- The number of tuples deleted depends on the number of tuples in the relation that satisfy the WHERE-clause
- Referential integrity should be enforced
DELETE (cont.)

Examples:
U4A: DELETE FROM EMPLOYEE
    WHERE LNAME='Brown'

U4B: DELETE FROM EMPLOYEE
    WHERE SSN='123456789'

U4C: DELETE FROM EMPLOYEE
    WHERE DNO IN
    (SELECT DNUMBER
     FROM DEPARTMENT
     WHERE DEPARTMENTNAME='Research')

U4D: DELETE FROM EMPLOYEE
UPDATE

- Used to modify attribute values of one or more selected tuples
- A WHERE-clause selects the tuples to be modified
- An additional SET-clause specifies the attributes to be modified and their new values
- Each command modifies tuples *in the same relation*
- Referential integrity should be enforced
Example: Change the location and controlling department number of project number 10 to 'Bellaire' and 5, respectively.

U5: UPDATE PROJECT
   SET PLOCATION = 'Bellaire', DNUM = 5
   WHERE NUMBER=10
Example: Give all employees in the 'Research' department a 10% raise in salary.

U6: UPDATE EMPLOYEE
    SET SALARY = SALARY * 1.1
    WHERE DNO IN (SELECT DNUMBER
                   FROM DEPARTMENT
                   WHERE DNAME='Research')

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

The reference to the SALARY attribute on the right of = refers to the old SALARY value before modification.

The reference to the SALARY attribute on the left of = refers to the new SALARY value after modification.
View Definition

- A view is a virtual relation that is derived from the relations in the conceptual schema or other existing views.
- A view does NOT exist in the physical form.
- A view can be used in queries as any "ordinary" relation. When it is used in a SELECT query, the virtual relation is computed first.

```sql
CREATE VIEW Emp_Activity
AS
(SELECT fname, lname, pname, hours
 FROM employee, works_on, project
 WHERE ssn = essn
 AND pno = pnumber
)

SELECT *
FROM Emp_Activity

DROP view Emp_Activity;
```