Chapter 5 Outline

- More Complex SQL Retrieval Queries
- Views (Virtual Tables) in SQL

Comparisons Involving NULL and Three-Valued Logic

- Meanings of NULL
  - Unknown value
  - Unavailable or withheld value
  - Not applicable attribute
- Each individual NULL value considered to be different from every other NULL value
- SQL uses a three-valued logic:
  - TRUE, FALSE, and UNKNOWN
Comparisons Involving NULL and Three-Valued Logic (cont’d.)

- SQL allows queries that check whether an attribute value is NULL
  - IS or IS NOT NULL

Query 18: Retrieve the names of all employees who do not have supervisors.

Q18: SELECT Name, Lastname
    FROM EMPLOYEE
    WHERE Super_sen IS NULL;

Nested Queries, Tuples, and Set/Multiset Comparisons

- Nested queries
  - Complete select-from-where blocks within WHERE clause of another query
  - Outer query
- Comparison operator IN
  - Compares value \( v \) with a set (or multiset) of values \( V \)
  - Evaluates to \( \text{TRUE} \) if \( v \) is one of the elements in \( V \)

Nested Queries (cont’d.)

- What does this say?

```
Q1A: SELECT DISTINCT PNum
    FROM PROJECT
    WHERE PNum IN
        ( SELECT PNum
          FROM PROJECT, DEPARTMENT, EMPLOYEE
          WHERE Dept=Dept/ AND
            Mgr_sen=Sen AND Lname='Smith' )
    OR
    PNum IN
        ( SELECT PNum
          FROM WORKS_ON, EMPLOYEE
          WHERE E=Sen AND Lname='Smith' );
```
Nested Queries (cont’d.)

- Use tuples of values in comparisons
  - Place them within parentheses
- What does this say?

```
SELECT DISTINCT E.name
FROM WORKS_ON
WHERE (Pho, Hours) IN (SELECT Pho, Hours
                        FROM WORKS_ON
                        Esn='123456789');
```

Nested Queries (cont’d.)

- Use other comparison operators to compare a single value \( v \)
  - \( \text{ANY (or = SOME)} \) operator
    - Returns TRUE if the value \( v \) is equal to some value in the set \( V \) and is hence equivalent to \( \text{IN} \)
  - Other operators that can be combined with \( \text{ANY (or SOME)} \): \( >, >=, <, <=, \) and \( <> \)

```
SELECT Name, FName
FROM EMPLOYEE
WHERE Salary > ALL (SELECT Salary
                    FROM EMPLOYEE
                    Dcn=3);
```

Nested Queries (cont’d.)

- Avoid potential errors and ambiguities
  - Create tuple variables (aliases) for all tables referenced in SQL query

```
Query 16. Retrieve the name of each employee who has a dependent with the same first name and is the same set as the employee.

One: SELECT E.FName, E.Lname
     FROM EMPLOYEE AS E
     WHERE E.Ssn IN (SELECT Dependent AS D
                        FROM E.FName=D.Dependent_name
                        AND E.Ssn=D.SSn);
```
**Correlated Nested Queries**

- **Correlated nested query**
  - Nested query references attribute in outer query
  - Nested with = or IN expressed as a single query

```sql
Select E.Fname, E.Lname
From Employee AS E, Dependent AS D
Where E.Ssn = D.Essn AND E.Sex = D.Sex AND E.Fname = D.Dependent_name
```

**The EXISTS and UNIQUE Functions in SQL**

- **EXISTS function**
  - Check whether the result of a correlated nested query is empty or not
- **EXISTS and NOT EXISTS**
  - Typically used in conjunction with a correlated nested query
- **SQL function UNIQUE(Q)**
  - Returns TRUE if there are no duplicate tuples in the result of query Q

**The EXISTS and UNIQUE Functions in SQL**

- Retrieve the names of employees who have no dependants

```sql
Select Fname, Lname
From Employee
Where Not Exists (Select *
  From Dependent
  Where Ssn = Essn)
```
Explicit Sets and Renaming of Attributes in SQL

- Can use explicit set of values in WHERE clause
- Use qualifier AS followed by desired new name
  - Rename any attribute that appears in the result of a query

```
SELECT E.Lname AS Employee_name, S.Lname AS Supervisor_name
FROM Employee AS E, Employee AS S
WHERE E.Supervisor = S.Employee;
```

Joined Tables in SQL

- Joined table
  - Permits users to specify a table resulting from a join operation in the FROM clause of a query
- Retrieve the name and address of all employees who work for the Research dept.

```
SELECT Fname, Lname, Address
FROM Employee
NATURAL JOIN (Department AS Dept(Dname, Dno, Msnn, Msdate))
WHERE Dname = 'Research';
```

Joined Tables in SQL

- NATURAL JOIN on two relations R and S
  - No join condition specified
  - Implicit EQUIJOIN condition for each pair of attributes with same name from R and S

```
Select Fname, Lname, Address
From Employee
Natural Join
(Dependent AS Dept(Dname, Dno, Msnn, Msdate))
Where Dname = 'Research';
```
Aggregate Functions in SQL

- Used to summarize information from multiple tuples into a single-tuple summary
- **Grouping**
  - Create subgroups of tuples before summarizing
- **Built-in aggregate functions**
  - COUNT, SUM, MAX, MIN, and AVG
- Functions can be used in the SELECT clause or in a HAVING clause

Aggregate Functions in SQL (cont'd.)

- NULL values discarded when aggregate functions are applied to a particular column

Query 20. Find the sum of the salaries of all employees of the 'Research' department, as well as the maximum salary, the minimum salary, and the average salary in this department.
Q20: SELECT SUM(Salary), MAX(Salary), MIN(Salary), AVG(Salary) 
FROM (EMPLOYEES JOIN DEPARTMENT ON Emp-DNumber) 
WHERE Department='Research';

Queries 21 and 22. Retrieve the total number of employees in the company (Q21) and the number of employees in the 'Research' department (Q22).
Q21: SELECT COUNT(*) 
FROM EMPLOYEE;
Q22: SELECT COUNT(*) 
FROM EMPLOYEE, DEPARTMENT 
WHERE DNumb=Emp-DNumber AND Department='Research';

Grouping: The GROUP BY and HAVING Clauses

- Partition relation into subsets of tuples
  - Based on grouping attribute(s)
  - Apply function to each such group independently
- **GROUP BY clause**
  - Specifies grouping attributes
- If NULLs exist in grouping attribute
  - Separate group created for all tuples with a NULL value in grouping attribute
Grouping: The GROUP BY and HAVING Clauses (cont’d.)

- **HAVING clause**
  - Provides a condition on the summary information

Query 2B. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than $40,000.

Q2B:  
```
SELECT Dnumber, COUNT(*)
FROM DEPARTMENT, EMPLOYEE
WHERE Dnumber=Dep AND Salary>40000 AND
(SELECT Dno
FROM EMPLOYEE
GROUP BY Dno
HAVING COUNT(*) > 5)
```  

Views (Virtual Tables) in SQL

- **Concept of a view in SQL**
  - Single table derived from other tables
  - Considered to be a virtual table

Specification of Views in SQL

- **CREATE VIEW command**
  - Give table name, list of attribute names, and a query to specify the contents of the view

V1:  
```
CREATE VIEW WORKS_ON1
AS SELECT Fname, Lname, Name, Hours
FROM EMPLOYEE, PROJECT, WORKS_ON
WHERE En--Ere AND Pro--Pnumb;
```  

V2:  
```
CREATE VIEW DEPT_INFO(Dep_name, No_of_ems, Total_sal)
AS SELECT Dname, COUNT(*), SUM(Salary)
FROM DEPARTMENT, EMPLOYEE
WHERE Dnumber=Dep
GROUP BY Dname;
```
Project SQL Queries...

- Get into your project group and determine SQL queries for the questions your application must answer.
- Will you need views, explain why or why not.