**Basic Retrieval Queries in SQL**

SQL has one basic statement for retrieving information from a database is the SELECT statement.

The SELECT-FROM-WHERE Structure of Basic SQL Queries

The basic form of the SELECT statement is

SELECT <attribute list>

FROM <table list>

WHERE <condition>;

where

■<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.

In SQL, the basic logical comparison operators for comparing attribute values with one another and with literal constants are =, <, <=, >, >=, and <>.

Retrieve the birth date and address of the employee(s) whose name is ‘John B. Smith’.

Q0: SELECT Bdate, Address

FROM EMPLOYEE

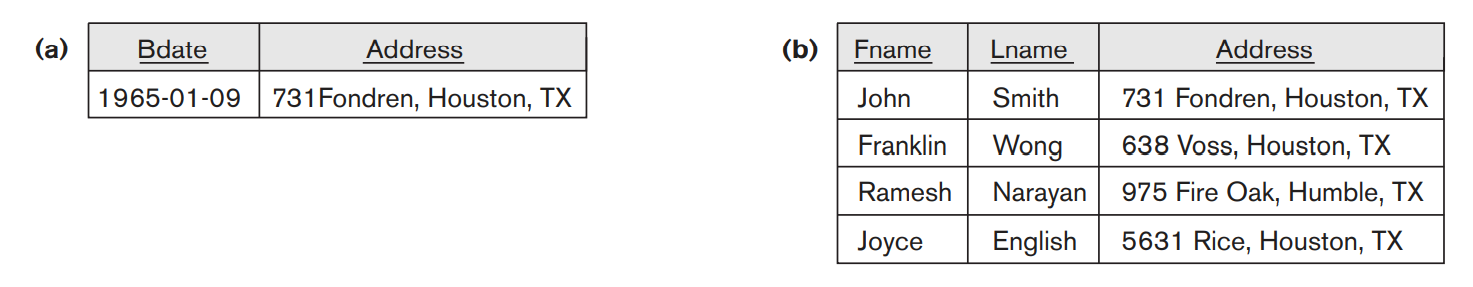
WHERE Fname = ‘John’ AND Minit = ‘B’ AND Lname = ‘Smith’;

Retrieve the name and address of all employees who work for the ‘Research’ department.

Q1: SELECT Fname, Lname, Address

FROM EMPLOYEE, DEPARTMENT

WHERE Dname = ‘Research’ AND Dnumber = Dno;



**Ambiguous Attribute Names, Aliasing, Renaming, and Tuple Variables**

In SQL, the same name can be used for two (or more) attributes as long as the attributes are in different tables. If this is the case, and a multi table query refers to two or more attributes with the same name, we must qualify the attribute name with the relation’s name to prevent ambiguity. This is done by prefixing the relation name to the attribute name and separating the two by a period.

SELECT EMPLOYEE.Fname, EMPLOYEE.LName,EMPLOYEE.Address

FROM EMPLOYEE, DEPARTMENT

WHERE DEPARTMENT.DName = ‘Research’ AND DEPARTMENT.Dnumber = EMPLOYEE.Dno;

We can use this alias-naming or renaming mechanism in any SQL query to specify tuple variables for every table in the WHERE clause, whether or not the same relation needs to be referenced more than once.

SELECT E.Fname, E.LName, E.Address

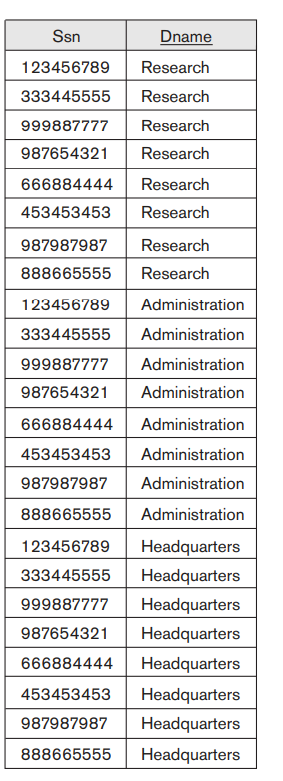
FROM EMPLOYEE AS E, DEPARTMENT AS D

WHERE D.DName = ‘Research’ AND D.Dnumber = E.Dno;

**Unspecified WHERE Clause and Use of the Asterisk**

A missing WHERE clause indicates no condition on tuple selection; hence, all tuples of the relation specified in the FROM clause qualify and are selected for the query result. If more than one relation is specified in the FROM clause and there is no WHERE clause, then the CROSS PRODUCT—all possible tuple combinations—of these relations is selected.

SELECT Ssn, Dname FROM EMPLOYEE, DEPARTMENT;



To retrieve all the attribute values of the selected tuples, we do not have to list the attribute names explicitly in SQL; we just specify an asterisk (\*), which stands for all the attributes.

SELECT \*

FROM EMPLOYEE

WHERE Dno = 5;

**Substring Pattern Matching and Arithmetic Operators**

The first feature allows comparison conditions on only parts of a character string, using the LIKE comparison operator.

This can be used for string pattern matching. Partial strings are specified using two reserved characters: % replaces an arbitrary number of zero or more characters, and the underscore (\_) replaces a single character.

Retrieve all employees whose address is in Houston, Texas.

SELECT Fname, Lname

FROM EMPLOYEE

WHERE Address LIKE ‘%Houston,TX%’;

Retrieve all employees in department 5 whose salary is between$30,000 and $40,000.

SELECT \*

FROM EMPLOYEE

WHERE (Salary BETWEEN 30000 AND 40000) AND Dno = 5;

**Ordering of Query Results**

SQL allows the user to order the tuples in the result of a query by the values of one or more of the attributes that appear in the query result, by using the ORDER BY clause.

The default order is in ascending order of values. We can specify the keyword DESC if we want to see the result in a descending order of values. The keyword ASC can be used to specify ascending order explicitly.

SELECT <attribute list>

FROM <table list>

[ WHERE <condition> ]

[ ORDER BY <attribute list> ];

Retrieve all employees in department 5 with fname as in ascending order

Select fname, lname from employee where dno=5 order by fname;