Basic SQL

Outline

- SQL Data Definition and Data Types
- Specifying Constraints in SQL
- Basic Retrieval Queries in SQL
- `INSERT`, `DELETE`, and `UPDATE` Statements in SQL
- Additional Features of SQL

Introduction to SQL

- “S.Q.L.” or “sequel”
- Supported by all major commercial database systems
- Standardized – many new features over time
- Interactive via GUI or prompt, or embedded in programs
- Declarative, based on relational algebra

SQL: Language Breakdown

- Data Definition Language (DDL)

  Create table…

  Alter table…

  Drop table…

- **Data Manipulation Language (DML)**

  C – INSERT/Create

  R – SELECT/ Reterieve

  U – UPDATE

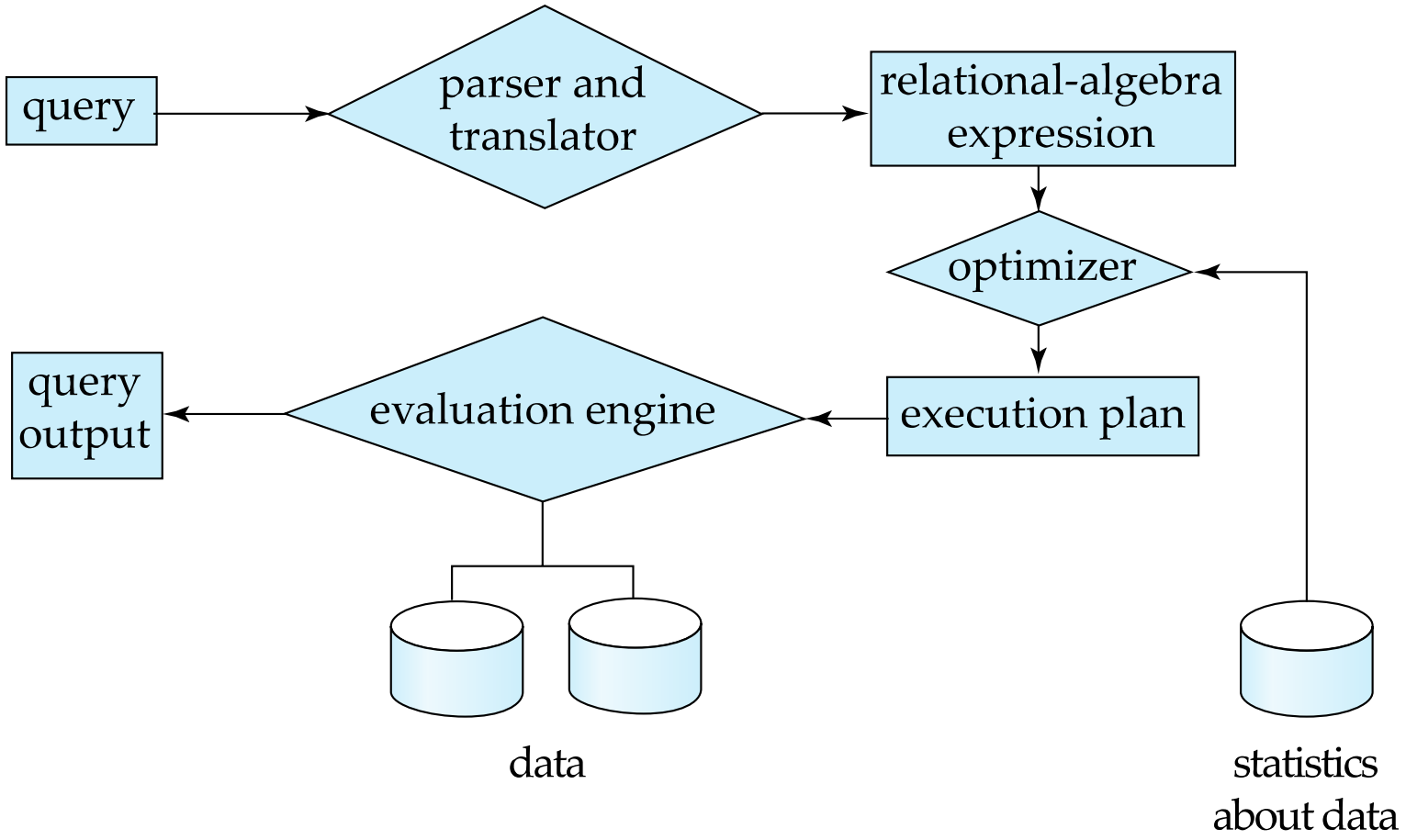
  D - DELETE

Other Commands

indexes, constraints, views, triggers, transactions, authorization, …

Basic Steps in Query Processing

1. Parsing and translation
2. Optimization
3. Evaluation



The Basic SELECT Statement

```
Select  A1,A2,…,An
From    R1,R2, …,Rm
Where   condition
```

What to return

Relations

Combine, filter

Relational Algebra Expression:

$\prod_{a1,a2,..an}(\delta_{condition} (R_1, R_2, \ldots, R_m))$

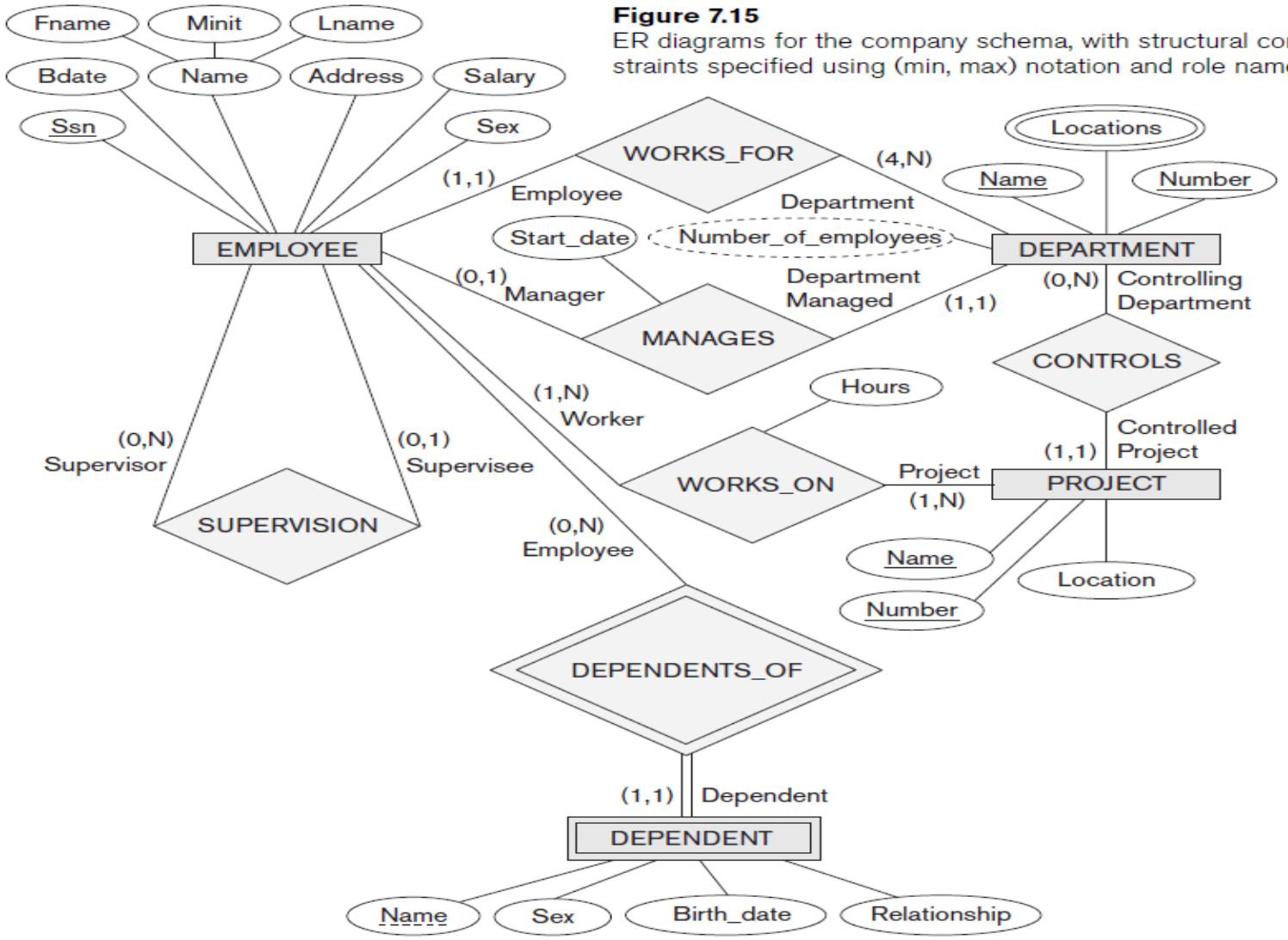


Figure 7.15
ER diagrams for the company schema, with structural constraints specified using (min, max) notation and role names.

Create Command

- Creating database

  CREATE DATABASE db_name

  Create database exam

- Next step is to create tables
- Two approaches:
  - Through SQL Create command
  - Through Enterprise Manager

Create Table Command

- Create table command is used to:
  - Create a table
  - Define attributes of the table with data types
  - Define different constraints on attributes, like primary and foreign keys, check constraint, not null, default value etc.

CREATE TABLE Construct

- CREATE TABLE [*schema*.] *name*

  (*column-1 TYPE* [DEFAULT *value*] [*constraints*],

  column-2 TYPE [DEFAULT *value*] [*constraints*],
  column-n TYPE [DEFAULT *value*] [*constraints*],

  [*table-constraints*]);

- Where:
  - [*schema*.]: **schema name followed by a dot.**
  - *name*: **table name**.
  - *column-1* to *column-n*: **column names**.
  - *TYPE*: **data type**.
  - [DEFAULT *value*]: **optional default value**.
  - [*constraints*]: **optional constraints, can be specified at column level or at table level**.

CREATE TABLE Example

```
CREATE TABLE EMPLOYEE
 (Fname          VARCHAR(15)          NOT NULL,
  Minit          CHAR,
  Lname          VARCHAR(15)          NOT NULL,
  Ssn            CHAR(9)              NOT NULL,
  Bdate          DATE,
  Address        VARCHAR(30),
  Sex            CHAR                 CHECK (LOWER(SEX) IN ('m',
'f')),
  Salary         DECIMAL(10,2),
  Super_ssn      CHAR(9),
  Dno            INT                  NOT NULL,
  CONSTRAINT EMPPK PRIMARY KEY (Ssn),
  CONSTRAINT EMPSUPERFK FOREIGN KEY (Super_ssn) REFERENCES
  EMPLOYEE (Ssn) ON DELETE SET NULL,
  CONSTRAINT EMPDEPTFK FOREIGN KEY (Dno) REFERENCES
  DEPARTMENT (Dnumber) ON DELETE SET NULL);
```

Creating tables for company database

```
CREATE TABLE EMPLOYEE
        ( Fname                 VARCHAR(15)             NOT NULL,
          Minit                 CHAR,
          Lname                 VARCHAR(15)             NOT NULL,
          Ssn                   CHAR(9)                 NOT NULL,
          Bdate                 DATE,
          Address               VARCHAR(30),
          Sex                   CHAR,
          Salary                DECIMAL(10,2),
          Super_ssn             CHAR(9),
          Dno                   INT                     NOT NULL,
        PRIMARY KEY (Ssn),
        FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn),
        FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber) );
CREATE TABLE DEPARTMENT
        ( Dname                 VARCHAR(15)             NOT NULL,
          Dnumber               INT                     NOT NULL,
          Mgr_ssn               CHAR(9)                 NOT NULL,
          Mgr_start_date        DATE,
        PRIMARY KEY (Dnumber),
        UNIQUE (Dname),
        FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn) );
```

```
CREATE TABLE DEPT_LOCATIONS
        ( Dnumber           INT             NOT NULL,
          Dlocation         VARCHAR(15)     NOT NULL,
        PRIMARY KEY (Dnumber, Dlocation),
        FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) );
CREATE TABLE PROJECT
        ( Pname             VARCHAR(15)     NOT NULL,
          Pnumber           INT             NOT NULL,
          Plocation         VARCHAR(15),
          Dnum              INT             NOT NULL,
        PRIMARY KEY (Pnumber),
        UNIQUE (Pname),
        FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber) );
CREATE TABLE WORKS_ON
        ( Essn              CHAR(9)         NOT NULL,
          Pno               INT             NOT NULL,
          Hours             DECIMAL(3,1)    NOT NULL,
        PRIMARY KEY (Essn, Pno),
        FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
        FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber) );
CREATE TABLE DEPENDENT
        ( Essn              CHAR(9)         NOT NULL,
          Dependent_name    VARCHAR(15)     NOT NULL,
          Sex               CHAR,
          Bdate             DATE,
          Relationship      VARCHAR(8),
        PRIMARY KEY (Essn, Dependent_name),
        FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn) );
```

Basic Retrieval Queries in SQL

- Basic structure of SQL consist of three clauses:
- `SELECT`: Correspond to the *projection* operation of RA
  - List the attributes desired in result of query
- `FROM`: Correspond to *Cartesian-product* of RA
  - List the relation to be scanned in the evaluation of the expression
- `WHERE`: Correspond to *Selection* of RA
  - Specify Boolean condition on attribute of relation appear on FROM clause that must be true for any retrieved tuple
  -

Select Examples

- Retrieve the *birth date* and *address* of the employee(s) whose name is '*John B. Smith*'

```
SELECT Bdate, Address
FROM EMPLOYEE
WHERE Fname='John' AND Minit='B' AND
Lname='Smith';
```

(a)

Bdate	Address
1965-01-09	731Fondren, Houston, TX

Select Examples

- Retrieve the name and address of all employees who work for the 'Research' department.

SELECT Fname, Lname, Address

FROM EMPLOYEE, DEPARTMENT

WHERE Dname='Research' AND Dnumber=Dno;

Fname	Lname	Address
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, Houston, TX
Ramesh	Narayan	975 Fire Oak, Humble, TX
Joyce	English	5631 Rice, Houston, TX

Unspecified WHERE Clause

- Missing `WHERE` clause
  - Indicates no condition on tuple selection
- 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.
- Following query Select all combinations of EMPLOYEE Ssn and DEPARTMENT Dname

SELECT Ssn, Dname
FROM EMPLOYEE, DEPARTMENT;

Select Distinct

- SQL does not automatically eliminate duplicate tuples in query results
  - Duplicate elimination is an expensive operation.

- Use the keyword **`DISTINCT`** in the `SELECT` clause
  - Only distinct tuples should remain in the result
- Retrieve the salary of every employee and all distinct salary value

SELECT ALL Salary

FROM EMPLOYEE;

SELECT DISTINCT Salary

FROM EMPLOYEE;

(a)

Salary
30000
40000
25000
43000
38000
25000
25000
55000

(b)

Salary
30000
40000
25000
43000
38000
55000

Substring Pattern Matching

- **`LIKE`** comparison operator
  - Used for string **pattern matching**
  - % replaces an arbitrary number of zero or more characters
- Retrieve all employees whose address is in Houston, Texas.

```
SELECT Fname, Lname
FROM EMPLOYEE
WHERE Address LIKE '%Houston%';
```

Substring Pattern Matching

- underscore (_) replaces a single character

```
SELECT Fname, Lname
FROM EMPLOYEE
WHERE Fname LIKE '_a%';
```

	FNAME	LNAME
1	Rameshn	Naraan
2	James	Borg

BETWEEN Operator

- **BETWEEN:** Checks the value in a range
- 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;
```

	fname	minit	lname	ssn	bdate	address	sex	salary	superssn	dno
1	Franklin	T	Wang	333445555	1955-12-08	638 Voss, Houston, TX	M	40000.00	888665555	5

IN Operator

- **IN**: Checks in a list of values

```
SELECT *
FROM EMPLOYEE
WHERE Salary IN (30000 ,40000) ;
```

	fname	minit	lname	ssn	bdate	address	sex	salary	superssn	dno
1	Justin	NULL	Mark	111111102	1966-01-12	2342 May, Atlanta, GA	M	40000.00	111111100	6
2	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000.00	333445555	NULL
3	Franklin	T	Wang	333445555	1955-12-08	638 Voss, Houston, TX	M	40000.00	888665555	5

`ORDER BY` clause

- Sorts the rows in a particular order

SELECT *select_list*

FROM *table_source*

WHERE *search_condition*

ORDER BY *order_expression* [ASC | DESC]

ORDER BY clause

- Retrieve a list of employees and the projects they are working on, ordered by *department* and, within each department, ordered alphabetically by last name, then first name.

SELECT D.Dname, E.Lname, E.Fname, P.Pname
FROM DEPARTMENT D, EMPLOYEE E, WORKS_ON W, PROJECT P
WHERE D.Dnumber= E.Dno **AND** E.Ssn= W.Essn **AND** W.Pno= P.Pnumber
ORDER BY D.Dname **Desc/Asc**, E.Lname, E.Fname;

Discussion and Summary of Basic SQL Retrieval Queries

- A simple retrieval query in SQL can consist of up to four clauses, but only the first two—SELECT and FROM—are mandatory.

```
SELECT       <attribute list>
FROM         <table list>
[ WHERE      <condition> ]
[ ORDER BY <attribute list> ];
```

INSERT, DELETE, and UPDATE Statements in SQL

- Three commands used to modify the database:
  - `INSERT`, `DELETE`, and `UPDATE`

The INSERT Command

- Add tuples to a relation.
  - Specify the relation name and a list of values for the tuple.

INSERT INTO EMPLOYEE

VALUES ('Richard', 'K', 'Marini', '653298653', '1962-12-30', '98 Oak Forest, Katy, TX', 'M', 37000, '653298653', 4);

- A second form of the INSERT statement allows the user to specify explicit attribute names that correspond to the values provided in the INSERT command.

INSERT INTO EMPLOYEE (Fname, Lname, Dno, Ssn)

VALUES ('Richard', 'Marini', 4, '653298653');

The DELETE Command

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

```
U4A:  DELETE FROM   EMPLOYEE
      WHERE         Lname='Brown';

U4B:  DELETE FROM   EMPLOYEE
      WHERE         Ssn='123456789';

U4C:  DELETE FROM   EMPLOYEE
      WHERE         Dno=5;

U4D:  DELETE FROM   EMPLOYEE;
```

The UPDATE Command

- Modify attribute values of one or more selected tuples
- Additional **`SET`** clause in the `UPDATE` command
  - Specifies attributes to be modified and new values

```
U5:   UPDATE   PROJECT
      SET      Plocation = 'Bellaire', Dnum = 5
      WHERE    Pnumber=10;
```

Additional Features of SQL

- Techniques for specifying complex retrieval queries
- Writing programs in various programming languages that include SQL statements
- Set of commands for specifying physical database design parameters, file structures for relations, and access paths
- Transaction control commands

Additional Features of SQL (cont'd.)

- Specifying the granting and revoking of privileges to users
- Constructs for creating triggers
- Enhanced relational systems known as object-relational
- New technologies such as XML and OLAP

Summary

- SQL
  - Comprehensive language
  - Data definition, queries, updates, constraint specification, and view definition
- Covered:
  - Data definition commands for creating tables
  - Commands for constraint specification
  - Simple retrieval queries
  - Database update commands