COSC 3380
Design of Database Systems

Data Modeling with Entity-Relationship (ER) Model

February 26, 2024

Operations on Relations

- INSERT a tuple
- MODIFY a tuple
- DELETE a tuple
- Integrity constraints should not be violated by these update operations.

Possible violations on INSERT

- INSERT may violate any of the constraints:
  - **Domain constraint**
  - **Key constraint**
  - **Referential integrity**
  - **Entity integrity**

Possible violations on DELETE

- DELETE may violate referential integrity:
  - If the primary key value of the tuple being deleted is referenced from other tuples in the database
    - Can be remedied by several actions: RESTRICT, CASCADE, SET NULL
      - RESTRICT option: reject the deletion
      - CASCADE option: propagate the new primary key value into the foreign keys of the referencing tuples
      - SET NULL option: set the foreign keys of the referencing tuples to NULL
  - One of the above options must be specified during database design for each foreign key constraint

Possible violations on DELETE

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

- Delete the WORKS_ON tuple with Essn = ‘999887777’ and Pno = 10
- Delete the EMPLOYEE tuple with Ssn = ‘999887777’
- Delete the EMPLOYEE tuple with Ssn = ‘333445555’

Possible violations on UPDATE/Modify

- UPDATE may violate domain constraint and NOT NULL constraint on an attribute being modified
- Any of the other constraints may also be violated, depending on the attribute being updated:
  - Updating the primary key (PK):
    - Similar to a DELETE followed by an INSERT
    - Need to specify similar options as DELETE
  - Updating a foreign key (FK):
    - May violate referential integrity
  - Updating an ordinary attribute (neither PK nor FK):
    - Can only violate domain constraints

Update Operations

- In case of integrity violation, several actions can be taken:
  - Cancel the operation that causes the violation (RESTRICT or REJECT option)
  - Perform the operation but inform the user of the violation
  - Trigger additional updates so the violation is corrected (CASCADE option, SET NULL option)
  - Execute a user-specified error-correction routine

Possible violations on UPDATE

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

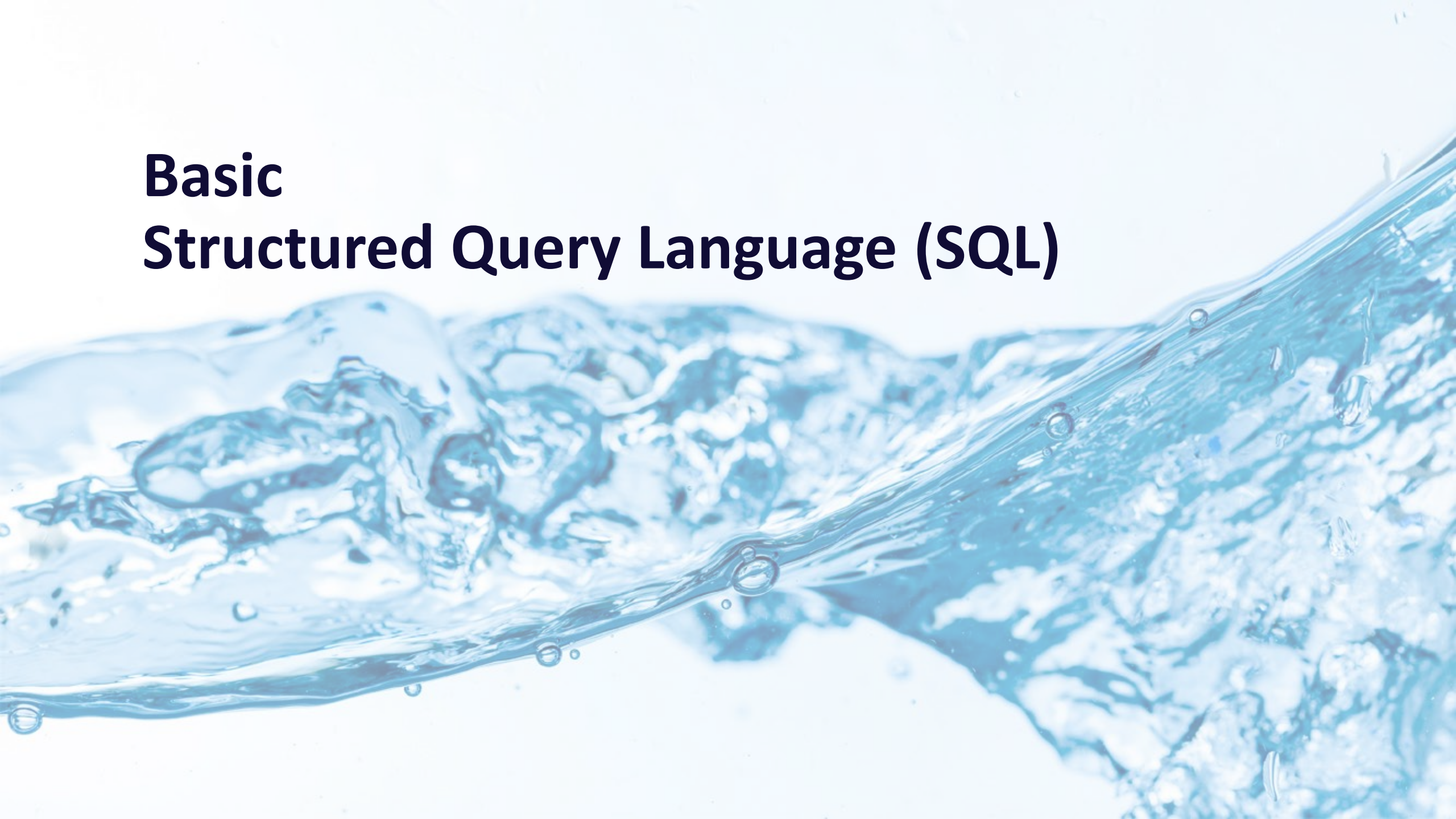
Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

- Update the salary of the EMPLOYEE tuple with Ssn = ‘999887777’ to 28000
- Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 7
- Update the Ssn of the EMPLOYEE tuple with Ssn = ‘999887777’ to ‘987654321’
- Update the Dno of the EMPLOYEE tuple with Ssn = ‘999887777’ to 1.

Integrity Constraints and Transactions

- Transaction
  - A program that includes some database operations:
  - reading from the database
  - inserting data
  - deleting data
  - updates to the data in the database
- At the end of a transaction, the database must remain in a valid state
  - Satisfying all constraints specified in the schema

Basic
Structured Query Language (SQL)



Basic SQL

- **SQL language**
  - Considered one of the major reasons for the commercial success of relational databases
- **SQL**
  - **Structured Query Language**
  - Statements for data definitions, queries, and updates (both DDL and DML)
  - **Core specification**
  - Plus specialized **extensions**

Key Characteristics of SQL

- Set-oriented and declarative
- Free form language
- Case insensitive
- Can be used interactively from a command prompt or executed by a program

SQL Data Definition & Data Types

- Terminology
  - **Table**, **row**, and **column** used for relational model terms relation, tuple, and attribute
- CREATE statement
  - Main SQL command for data definition

Schema & Catalog Concepts

- **SQL schema**
  - Identified by a **schema name**
  - Includes an **authorization identifier** and **descriptors** for each element
- Schema **elements** include
  - Tables, constraints, views, domains, and other constructs
- Each statement in SQL ends with a semicolon

Schema and Catalog Concepts

- CREATE SCHEMA statement

  CREATE SCHEMA COMPANY AUTHORIZATION ‘Jsmith’;

  CREATE DATABASE COMPANY AUTHORIZATION ‘Jsmith’;

- **Catalog**
  - Named collection of schemas in an SQL environment
- **SQL environment**
  - Installation of an SQL-compliant RDBMS on a computer system

CREATE TABLE Command

- Specify a new relation
  - Provide name
  - Specify attributes and initial constraints
- Can optionally specify schema:
  - CREATE TABLE COMPANY.EMPLOYEE ...

    or
  - CREATE TABLE EMPLOYEE ...

CREATE TABLE Command

- **Base relations (base tables)**
  - Relation and its tuples are actually created and stored as a file by the DBMS
- **Virtual relations**
  - Created through the CREATE VIEW statement
  - Why do we need views?

Create Table Construct

- A relation is defined using the **create table** command:

  create table *r*

  (A_1 D_1, A_2 D_2, …, A_n D_n,
  (integrity-constraint$_1$), …, (integrity-constraint$_k$))

  - *r* is the name of the relation/table
  - each A_i is an attribute name in the schema of relation *r*
  - D_i is the data type of values in the domain of attribute A_i
- Example:

  create table *instructor* (
  ID **char**(5),
  name **varchar**(20),
  dept_name **varchar**(20),
  salary **numeric**(8,2))

Create Table Construct

- **create table** *student* (
  ID **varchar**(5),
  name **varchar**(20) not null,
  dept_name **varchar**(20),
  tot_cred **numeric**(3,0),
  primary key *(ID)*,
  foreign key *(dept_name*) **references** *department*);

- **create table** *takes* (
  ID **varchar**(5),
  course_id **varchar**(8),
  sec_id **varchar**(8),
  semester **varchar**(6),
  year **numeric**(4,0),
  grade **varchar**(2),
  primary key *(ID, course_id, sec_id, semester, year)* ,
  foreign key (*ID*) **references** *student*,
  foreign key (*course_id, sec_id, semester, year*) **references** *section*);

Create Table Construct

- **create table** *course* (
  course_id **varchar**(8),
  title **varchar(**50),
  dept_name **varchar**(20),
  credits **numeric**(2,0),
  primary key *(course_id),*
  foreign key *(dept_name*) **references** *department*);

```
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) );
```

DDL Concepts using SQL

Data Type	Description
CHAR(n)	Holds a fixed length string with size n
VARCHAR(n)	Holds a variable length string with maximum size n
SMALLINT	Small integer (no decimal) between -32768 to 32767
INT	Integer (no decimal) between -2147483648 to 2147483647
FLOAT(n,d)	Small number with a floating decimal point. The total maximum number of digits is n with a maximum of d digits to the right of the decimal point.
DOUBLE(n,d)	Large number with a floating decimal point. The total maximum number of digits is n with a maximum of d digits to the right of the decimal point.
DATE	Date in format YYYY-MM-DD
DATETIME	Date and time in format YYYY-MM-DD HH:MI:SS
TIME	Time in format HH:MI:SS
BOOLEAN	True or False
BLOB	Binary Large Object (e.g. image, audio, video)

DDL Concepts using SQL

```
CREATE TABLE EMPLOYEE
    ( … ,
      Dno          INT          NOT NULL       DEFAULT 1,
    CONSTRAINT EMPPK
      PRIMARY KEY (Ssn),
    CONSTRAINT EMPSUPERFK
      FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn)
                  ON DELETE SET NULL          ON UPDATE CASCADE,
    CONSTRAINT EMPDEPTFK
      FOREIGN KEY(Dno) REFERENCES DEPARTMENT(Dnumber)
                  ON DELETE SET DEFAULT       ON UPDATE CASCADE);
CREATE TABLE DEPARTMENT
    ( … ,
      Mgr_ssn CHAR(9)          NOT NULL       DEFAULT '888665555',
      … ,
    CONSTRAINT DEPTPK
      PRIMARY KEY(Dnumber),
    CONSTRAINT DEPTSK
      UNIQUE (Dname),
    CONSTRAINT DEPTMGRFK
      FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
                  ON DELETE SET DEFAULT       ON UPDATE CASCADE);
CREATE TABLE DEPT_LOCATIONS
    ( … ,
    PRIMARY KEY (Dnumber, Dlocation),
    FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber)
                  ON DELETE CASCADE           ON UPDATE CASCADE);
```

DDL Concepts using SQL

```
CREATE DOMAIN PRODTYPE_DOMAIN AS VARCHAR(10);
CREATE DOMAIN SSN_TYPE AS CHAR(9);
```

Additional constraints on individual tuples within a relation are also possible using **CHECK**

```
CHECK (VALUE IN ('white', 'red', 'rose', 'sparkling'))

Dnumber INT NOT NULL CHECK (Dnumber > 0 AND Dnumber < 21);
```

CHECK clauses at the end of a **CREATE TABLE** statement

- **CHECK** (Dept_create_date <= Mgr_start_date);

Apply to each tuple individually

DDL Concepts using SQL

- Column constraints
  - **PRIMARY KEY** constraint defines the primary key of the table
  - **FOREIGN KEY** constraint defines a foreign key of a table
  - **UNIQUE** constraint defines an alternative key of a table
  - **NOT NULL** constraint prohibits NULL values for a column
  - **DEFAULT** constraint sets a default value for a column
  - **CHECK** constraint defines a constraint on the column values

DROP and ALTER Command

- **DROP** command can be used to drop or remove database objects
  - can be combined with CASCADE and RESTRICT
- Examples:

```
DROP SCHEMA PURCHASE CASCADE
DROP SCHEMA PURCHASE RESTRICT
DROP TABLE PRODUCT CASCADE
DROP TABLE PRODUCT RESTRICT
```

DROP and ALTER Command

- **ALTER** statement can be used to modify table column definitions
- Examples:

```
ALTER TABLE PRODUCT ADD PRODIMAGE BLOB
```

```
ALTER TABLE SUPPLIER ALTER SUPSTATUS SET DEFAULT '10'
```

SQL Data Manipulation Language (DML)

- SQL SELECT Statement
- SQL INSERT Statement
- SQL DELETE Statement
- SQL UPDATE Statement

SQL Select Statement

- Basic statement for retrieving information from a database
- Result of SQL SELECT statement is a multiset, and not a set!
- Multiset (bag behavior)
  - Elements are not ordered AND there can be duplicates
- Examples:
  - Set {10, 5, 20} Multiset {10, 5, 10, 20, 5, 10}
- SQL does not eliminate duplicates
  - duplicate elimination is expensive
  - user may want to see duplicate tuples
  - duplicates may be considered by aggregate functions
- Tuple-id may be used as a key

Basic SQL Queries: SELECT-FROM-WHERE Structure

```
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.