

Module 3

Relational Databases

SQL-99: Data Definition Language (DDL)

Data Definition, Constraints, and Schema Changes

- Used to CREATE, DROP, and ALTER the descriptions of the tables (relations) of a database

CREATE TABLE

- Specifies a new base relation by giving it a name, and specifying each of its attributes and their data types (NUMBER, CHAR(n), VARCHAR2(n))
- A constraint NOT NULL may be specified on an attribute

```
CREATE TABLE DEPARTMENT  
(  
    DNAME VARCHAR2(10) NOT NULL,  
    DNUMBER      NUMBER NOT NULL,  
    MGRSSN       CHAR(9),  
    MGRSTARTDATE CHAR(9)  
);
```

CREATE TABLE

- In SQL2, can use the CREATE TABLE command for specifying the primary key attributes, secondary keys, and referential integrity constraints (foreign keys).
- Key attributes can be specified via the PRIMARY KEY and UNIQUE phrases

```
CREATE TABLE DEPTARTMENT  
( DNUMBER NUMBER PRIMARY KEY,  
  DNAME VARCHAR2(10) NOT NULL,  
  MGRSSN CHAR(9) REFERENCES EMP,  
  MGRSTARTDATE CHAR(9),  
  CONSTRAINT DEPT_DNAME_UK UNIQUE(DNAME)  
)
```

CREATE TABLE

- In SQL2, can use the CREATE TABLE command for specifying the primary key attributes, secondary keys, and referential integrity constraints (foreign keys).
- Key attributes can be specified via the PRIMARY KEY and UNIQUE phrases

```
CREATE TABLE DEPT
( DNUMBER NUMBER,
  DNAME VARCHAR2(10) NOT NULL,
  MGRSSN CHAR(9),
  MGRSTARTDATE CHAR(9),
  CONSTRAINT DEPT_DNAME_UK UNIQUE(DNAME),
  CONSTRAINT DEPT_MGRNSS_FK FOREIGN KEY (MGRSSN) REFERENCES EMP,
  CONSTRAINT DEPT_DNUMBER_PK PRIMARY KEY (DNUMBER)
)
```

DROP TABLE

- Used to remove a relation (base table) *and its definition*
- The relation can no longer be used in queries, updates, or any other commands since its description no longer exists
- Example:

DROP TABLE DEPENDENT;

You cannot drop a table if there is a referential integrity constraints which refer to primary key in the table.

ALTER TABLE

- Used to add an attribute to one of the base relations
- The new attribute will have NULLs in all the tuples of the relation right after the command is executed; hence, the NOT NULL constraint is *not allowed* for such an attribute (if the table already contains data).
- Example:

ALTER TABLE EMPLOYEE ADD JOB VARCHAR2(12);

- The database users must still enter a value for the new attribute JOB for each EMPLOYEE tuple. This can be done using the UPDATE command.

ALTER TABLE EMPLOYEE MODIFY (JOB NOT NULL);

ALTER TABLE

- Also used to add constraints:
- Examples:

```
ALTER TABLE DEPT ADD CONSTRAINT DEPT_DNAME_UK UNIQUE(DNAME);
```

- Current data in the table (if exists) cannot violate the new constraint.

Features Added in SQL2 and SQL-99

- **REFERENTIAL INTEGRITY OPTIONS**

REFERENTIAL INTEGRITY OPTIONS

- We can specify RESTRICT, CASCADE, SET NULL or SET DEFAULT on referential integrity constraints (foreign keys)
 - Restrict: Disallows the update or deletion of referenced data.
 - Set to Null: When referenced data is updated or deleted, all associated dependent data is set to NULL.
 - Set to Default: When referenced data is updated or deleted, all associated dependent data is set to a default value.
 - Cascade: When referenced data is updated, all associated dependent data is correspondingly updated. When a referenced row is deleted, all associated dependent rows are deleted.

REFERENTIAL INTEGRITY OPTIONS

- We can specify RESTRICT, CASCADE, SET NULL or SET DEFAULT on referential integrity constraints (foreign keys)

```
CREATE TABLE DEPT  
(  
    DNAME          VARCHAR(10) NOT NULL,  
    DNUMBER        INTEGER      NOT NULL,  
    MGRSSN          CHAR(9),  
    MGRSTARTDATE    CHAR(9),  
    PRIMARY KEY (DNUMBER),  
    UNIQUE (DNAME),  
    FOREIGN KEY (MGRSSN) REFERENCES EMP  
ON DELETE SET NULL);
```

REFERENTIAL INTEGRITY OPTIONS (continued)

Oracle supports the use of FOREIGN KEY integrity constraints to define the referential integrity actions, including:

- Update and delete No Action
- Delete CASCADE
- Delete SET NULL

Additional Data Types in SQL2 and SQL-99

Has DATE AND TIME data types

- **DATE:**
 - Stores date and time information. For each DATE value, Oracle stores the following information: century, year, month, date, hour, minute, and second.
- **TIMESTAMP:**
 - It stores year, month, day, hour, minute, and second values. It also stores fractional seconds, which are not stored by the DATE datatype.
- **TIMESTAMP WITH TIME ZONE:**
 - Includes a time zone offset or time zone region name in its value.
- **TIMESTAMP WITH LOCAL TIME ZONE:**
 - data stored in the database is normalized to the database time zone, and the time zone offset is not stored as part of the column data.

Additional Data Types in SQL2 and SQL-99 (cont.)

- **INTERVAL:**

- Specifies a relative value rather than an absolute value
- Can be DAY TO SECOND intervals or YEAR TO MONTH intervals
 - INTERVAL YEAR TO MONTH – stores intervals using of year and month. (2 years and 3 months)
 - INTERVAL DAY TO SECOND – stores intervals using days, hours, minutes, and seconds including fractional seconds. (11 days, 10 hours, 09 minutes, 08 seconds, and 555 thousandths of a second)
- Can be positive or negative when added to or subtracted from an absolute value, the result is an absolute value

Specifying Updates in SQL

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

INSERT

- In its simplest form, 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

INSERT (cont.)

- Example:

**U1: INSERT INTO EMPLOYEE
VALUES ('Richard','K','Marini', '653298653', '30-DEC-52',
'98 Oak Forest,Katy,TX', 'M', 37000,'987654321', 4)**

- An alternate form of INSERT specifies explicitly the attribute names that correspond to the values in the new tuple
- Attributes with NULL values can be left out
- Example: Insert a tuple for a new EMPLOYEE for whom we only know the FNAME, LNAME, and SSN attributes.

**U1A: INSERT INTO EMPLOYEE (FNAME, LNAME, SSN)
VALUES ('Richard', 'Marini', '653298653')**

INSERT (cont.)

- Important Note: Only 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

INSERT (cont.)

- 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 VARCHAR2(10),
 NO_OF_EMPS NUMBER,
 TOTAL_SAL NUMBER);

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 ;

INSERT (cont.)

- 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 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 DNAME='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

UPDATE (cont.)

- 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	PNUMBER=10

UPDATE (cont.)

- 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