Relational Database Management System 116U01C403

Module 3 .2 Jan 2023-May 2023

Relational Algebra & SQL

Module 3.2

- Overview of SQL
- Data Definition Commands
- Domain Constraints
- Referential integrity

Overview of SQL

- Structured Query Language (SQL)
- The **Data Definition Language Commands** in Structured Query Language are used to create and modify the schema of the database and its objects.
- The **Data Manipulation Language Commands** in Structured Query Language (SQL) deal with the manipulation of data present in the database.

DDL commands

- CREATE Command.
- DROP Command.
- ALTER Command.
- TRUNCATE Command.
- RENAME Command.

Domain Types in SQL

- **char(n).** Fixed length character string, with user-specified length *n*.
- **varchar(n).** Variable length character strings, with user-specified maximum length *n*.
- int. Integer (a finite subset of the integers that is machine-dependent).
- smallint. Small integer (a machine-dependent subset of the integer domain type).
- **numeric(p,n).** Fixed point number, with user-specified precision of *p* digits, with *n* digits to the right of decimal point.
- **real, double precision.** Floating point and double-precision floating point numbers, with machine-dependent precision.
- float(n). Floating point number, with user-specified precision of at least *n* digits.

CREATE TABLE Command

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

CREATE TABLE DEPARTMENT (DNAME VARCHAR(10)NOT NULL, DNUMBER INTEGER NOT NULL, MGRSSN CHAR(9), MGRSTARTDATE CHAR(9));

CREATE TABLE Command

- In SQL2, CREATE TABLE command can be used 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 (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);

DROP TABLE Command

- 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;

ALTER TABLE Command

- 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
- Example:

ALTER TABLE EMPLOYEE ADD JOB VARCHAR(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.

DDL commands in SQL

- CREATE SCHEMA
- REFERENTIAL INTEGRITY OPTIONS

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 DEFAULT ON UPDATE CASCADE );
```

Referential Integrity Option

CREATE TABLE EMP (ENAME VARCHAR(30) NOT NULL, ESSN CHAR(9), BDATE DATE, DNO INTEGER DEFAULT 1, SUPERSSN CHAR(9), PRIMARY KEY (ESSN), FOREIGN KEY (DNO) REFERENCES DEPT ON DELETE SET DEFAULT ON UPDATE CASCADE, FOREIGN KEY (SUPERSSN) REFERENCES EMP ON DELETE SET NULL ON UPDATE CASCADE);

Additional Data Types

Has DATE, TIME, and TIMESTAMP data types

- DATE:
 - Made up of year-month-day in the format yyyy-mm-dd
- TIME:
 - Made up of hour:minute:second in the format hh:mm:ss
- TIME(i):
 - Made up of hour:minute:second plus i additional digits specifying fractions of a second
 - format is hh:mm:ss:ii...i

• TIMESTAMP:

• Has both DATE and TIME components

Additional Data

• INTERVAL:

- Specifies a relative value rather than an absolute value
- Can be DAY/TIME intervals or YEAR/MONTH intervals
- Can be positive or negative when added to or subtracted from an absolute value, the result is an absolute value

DML commands in SQL

- Used for accessing/ manipulating data
- SELECT Command.
- INSERT Command.
- UPDATE Command.
- DELETE Command.

Retrieval Queries in SQL

- SQL has one basic statement for retrieving information from a database; the SELECT statement
- This is *not the same* as the SELECT operation of the relational algebra
- Important distinction between SQL and the formal relational model; SQL allows a table (relation) to have two or more tuples that are identical in all their attribute values
- Hence, an SQL relation (table) is a *multi-set* (sometimes called a bag) of tuples; it *is not* a set of tuples
- SQL relations can be constrained to be sets by specifying PRIMARY KEY or UNIQUE attributes, or by using the DISTINCT option in a query

Retrieval Queries in SQL (cont.)

- Basic form of the SQL SELECT statement is called a *mapping* or a *SELECT*-*FROM-WHERE block*
 - SELECT<attribute list>FROMWHERE<condition>
 - <attribute list> is a list of attribute names whose values are to be retrieved by the query
 - 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

Relational Database Schema--Figure 5.5

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	-----	-------	---------	-----	--------	----------	-----

DEPARTMENT

DNAME <u>DNUMBER</u>	MGRSSN	MGRSTARTDATE
----------------------	--------	--------------

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM
-------	---------	-----------	------

WORKS_	ON
--------	----

|--|

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

							1			
EMPLOYEE	FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	В	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	М	30000	333445555	5
	Franklin	Т	Wong	333445555	1955-12-08	638 Voss, Houston, TX	М	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	К	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	М	38000	333445555	5
	Joyce	А	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	М	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	М	55000	null	1

					DEPT_LOCATIO	ONS	D
DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGF	RSTARTDATE		
	Research	5	333445555	1	988-05-22	[
	Administration	4	987654321	1	995-01-01	[
	Headquarters	1	888665555	1	981-06-19	-	

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston
	DNUMBER 1 4 5 5 5 5

Populated Database--Fig.5.6

WORKS_ON	<u>ESSN</u>	<u>PNO</u>	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

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	ProductX	1	Bellaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization Newbenefits		Houston	1
			Stafford	4

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	м	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	м	1942-02-28	SPOUSE
	123456789	Michael	м	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

Simple SQL Queries

- Basic SQL queries correspond to using the SELECT, PROJECT, and JOIN operations of the relational algebra
- All subsequent examples use the COMPANY database
- Example of a simple query on *one* relation
- <u>Query 0:</u> Retrieve the birthdate and address of the employee whose name is 'John B. Smith'.

Q0: SELECT	BDATE, ADDRESS
FROM	EMPLÓYEE
WHERE	FNAME='John' AND MINIT='B'
AND	LNAME='Smith'

- Similar to a SELECT-PROJECT pair of relational algebra operations; the SELECT-clause specifies the *projection attributes* and the WHERE-clause specifies the *selection condition*
- However, the result of the query *may contain* duplicate tuples

Simple SQL Queries (cont.)

• <u>Query 1:</u> Retrieve the name and address of all employees who work for the 'Research' department.

Q1:SELECTFNAME, LNAME, ADDRESSFROMEMPLOYEE, DEPARTMENTWHEREDNAME='Research' AND DNUMBER=DNO

- Similar to a SELECT-PROJECT-JOIN sequence of relational algebra operations
- (DNAME='Research') is a *selection condition* (corresponds to a SELECT operation in relational algebra)
- (DNUMBER=DNO) is a *join condition* (corresponds to a JOIN operation in relational algebra)

Simple SQL Queries (cont.)

• <u>Query 2:</u> For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate.

Q2: SELECTPNUMBER, DNUM, LNAME, BDATE, ADDRESSFROMPROJECT, DEPARTMENT, EMPLOYEEWHEREDNUM=DNUMBER AND MGRSSN=SSN AND PLOCATION='Stafford'

- In Q2, there are *two* join conditions
- The join condition DNUM=DNUMBER relates a project to its controlling department
- The join condition MGRSSN=SSN relates the controlling department to the employee who manages that department

Aliases, * and DISTINCT, Empty WHERE-clause

• In SQL, we can use the same name for two (or more) attributes as long as the attributes are in *different relations*

A query that refers to two or more attributes with the same name must *qualify* the attribute name with the relation name by *prefixing* the relation name to the attribute name

Example:

• EMPLOYEE.LNAME, DEPARTMENT.DNAME

ALIASES

- Some queries need to refer to the same relation twice
- In this case, *aliases* are given to the relation name
- <u>Query 8:</u> For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.

Q8: SELECTE.FNAME, E.LNAME, S.FNAME, S.LNAMEFROMEMPLOYEE E SWHEREE.SUPERSSN=S.SSN

- In Q8, the alternate relation names E and S are called *aliases* or *tuple variables* for the EMPLOYEE relation
- We can think of E and S as two *different copies* of EMPLOYEE; E represents employees in role of *supervisees* and S represents employees in role of *supervisors*

ALIASES (cont.)

- Aliasing can also be used in any SQL query for convenience Can also use the AS keyword to specify aliases
 - Q8: SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME FROM EMPLOYEE AS E, EMPLOYEE AS S WHERE E.SUPERSSN=S.SSN

UNSPECIFIED WHERE-clause

- A *missing WHERE-clause* indicates no condition; hence, *all tuples* of the relations in the FROM-clause are selected
- This is equivalent to the condition WHERE TRUE
- <u>Query 9:</u> Retrieve the SSN values for all employees.

Q9: SELECT SSN FROM EMPLOYEE

• If more than one relation is specified in the FROM-clause *and* there is no join condition, then the *CARTESIAN PRODUCT* of tuples is selected

UNSPECIFIED WHERE-clause (cont.)

• Example:

Q10:SELECTSSN, DNAMEFROMEMPLOYEE, DEPARTMENT

• It is extremely important not to overlook specifying any selection and join conditions in the WHERE-clause; otherwise, incorrect and very large relations may result

USE OF *

- To retrieve all the attribute values of the selected tuples, a * is used, which stands for *all the attributes* <u>Examples:</u>
 - Q1C:SELECT*
FROM
WHEREEMPLOYEE
DNO=5Q1D:SELECT*
FROM
WHEREPLOYEE, DEPARTMENT
WHEREDNAME='Research' AND DNO=DNUMBER

USE OF DISTINCT

- SQL does not treat a relation as a set; *duplicate tuples can appear*
- To eliminate duplicate tuples in a query result, the keyword **DISTINCT** is used
- For example, the result of Q11 may have duplicate SALARY values whereas Q11A does not have any duplicate values

Q11:SELECTSALARYFROMEMPLOYEEQ11A:SELECTDISTINCT SALARYFROMEMPLOYEE