

IT420: Database Management and Organization

SQL: Structured Query Language (Chapter 7)

1

Goals

- SQL: Data Definition Language
 - CREATE
 - ALTER
 - DROP
- SQL: Data Manipulation Language
 - INSERT
 - DELETE
 - UPDATE
 - SELECT

2

Relational Query Languages

- A major strength of the relational model:
 - supports simple, powerful querying of data
- Queries can be written intuitively, and the DBMS is responsible for efficient evaluation.

3

SQL DDL and DML

- SQL statements can be divided into two categories:
 - **Data definition language (DDL)** statements
 - Used for creating and modifying tables, views, and other structures
 - CREATE, DROP, ALTER
 - **Data manipulation language (DML)** statements.
 - Used for queries and data modification
 - INSERT, DELETE, UPDATE, SELECT

4

Creating Tables

```
CREATE TABLE table_name(  
    column_name1 column_type1 [constraints1],  
    .....,  
    [[CONSTRAINT constraint_name] table_constraint]  
)
```

Table constraints:

- NULL/NOT NULL
- PRIMARY KEY (*columns*)
- UNIQUE (*columns*)
- CHECK (*conditions*)
- FOREIGN KEY (*local_columns*) REFERENCES *foreign_table* (*foreign_columns*) [ON DELETE *action_d* ON UPDATE *action_u*]

Specify surrogate key in SQL Server:

```
column_name int_type IDENTITY (seed, increment)
```

5

CREATE TABLE Example

- CREATE TABLE Students
(StudentNumber integer NOT NULL,
StudentLastName char(18) NOT NULL,
StudentFirstName char(18) NOT NULL,
Email char(50),
PhoneNumber char(18),
MajorDepartmentName char(18),

CONSTRAINT PK_Students PRIMARY KEY (StudentNumber),
CONSTRAINT U_Email UNIQUE (Email),
CONSTRAINT FK_Dept FOREIGN KEY(MajorDepartmentName)
REFERENCES DEPARTMENTS(DepartmentName)
ON DELETE NO ACTION ON UPDATE CASCADE
)

6

FOREIGN KEY Constraints

DEPARTMENTS

- DepartmentName: char(18)
- Phone: char(18)
- Building: char(18)
- Room: integer

U:C
U:SN

Majors

I:SN
U:SN

STUDENTS

- StudentNumber: integer
- StudentLastName: char(18)
- StudentFirstName: char(18)
- Email: varchar(50)
- PhoneNumber: char(18)
- DepartmentName: char(18) (FK)

DepartmentName	Phone	Building	Room
Mathematics	410-293-4573	Michelson Hall	308
History	410-293-2265	Sampson Hall	120
Computer Science	410-293-6800	Michelson Hall	340

Student Number	Student LastName	Student FirstName	Email	PhoneNumber	MajorDepartmentName
190	Smith	John	jsmith@ucrna.edu	410-431-3456	
673	Doe	Jane	jdoe@ucrna.edu		Computer Science
312	Doe	Bob	bdoe@ucrna.edu	443-451-7865	Mathematics

CREATE TABLE Departments
(DepartmentName char(18),
Phone char(18) NOT NULL,
Building char(18),
Room integer,
PRIMARY KEY (DepartmentName)
)

FOREIGN KEY Constraints

- 4 options on deletes and updates:
 - NO ACTION (delete/update is rejected)
 - CASCADE
 - SET NULL
 - SET DEFAULT

```
CREATE TABLE Students
(StudentNumber integer,
StudentLastName char(18) NOT NULL,
StudentFirstName char(18) NOT NULL,
Email char(50) NULL,
PhoneNumber char(18) NULL,
MajorDepartmentName char(18) NULL,
PRIMARY KEY (StudentNumber),
UNIQUE(Email),
FOREIGN KEY (MajorDepartmentName)
REFERENCES Departments (DepartmentName)
ON DELETE SET NULL
ON UPDATE CASCADE
)
```

Modifying Tables

- ALTER TABLE *table_name* clause

Clauses:

ADD COLUMN *column_name* *column_type* [*constraints*]
 DROP COLUMN *column_name*
 ALTER COLUMN / MODIFY – DBMS specific!
 ADD CONSTRAINT *constraint*
 DROP CONSTRAINT *constraint_name*

ALTER TABLE Examples

- ALTER TABLE Students ADD COLUMN BirthDate datetime NULL
- ALTER TABLE Students DROP COLUMN BirthDate
- ALTER TABLE Student ADD CONSTRAINT FK_Department FOREIGN KEY (MajorDepartmentName) REFERENCES Departments (DepartmentName) ON DELETE NO ACTION ON UPDATE CASCADE

Removing Tables

- DROP TABLE *table_name*

DROP TABLE Departments;

- If there are constraints dependent on table:

- Remove constraints
- Drop table

ALTER TABLE Students

DROP CONSTRAINT FK_Department;

DROP TABLE Departments;

SQL DDL and DML

- Data definition language (DDL)** statements
 - Used for creating and modifying tables, views, and other structures
 - CREATE, ALTER, DROP
- Data manipulation language (DML)** statements.
 - Used for queries and data modification
 - INSERT, DELETE, UPDATE, SELECT

SQL DML

▪ Data manipulation language (DML) statements.

- Used for queries and data modification
- INSERT
- DELETE
- UPDATE
- SELECT

13

INSERT Statement

```
INSERT INTO table_name [ (column_list) ] VALUES (data_values)
INSERT INTO table_name [ (column_list) ] select_statement
```

INSERT command:

```
INSERT INTO Students (StudentNumber, StudentLastName, StudentFirstName)
VALUES (190, 'Smith', 'John');

INSERT INTO Students VALUES (190, 'Smith', 'John', 'jsmith@usna.edu',
'410-431-3456')
```

▪ Bulk INSERT:

```
INSERT INTO Students (StudentNumber, StudentLastName, StudentFirstName,
Email, PhoneNumber)
SELECT *
FROM Second_Class_Students;
```

14

UPDATE Statement

```
UPDATE table_name
SET column_name1 = expression1 [, column_name2 = expression2,... ]
[ WHERE search_condition ]
```

▪ UPDATE command:

```
UPDATE Students
SET PhoneNumber = '410-123-4567'
WHERE StudentNumber = 673;
```

▪ BULK UPDATE command:

```
UPDATE Students
SET PhoneNumber = '410-123-4567'
WHERE StudentLastName = 'Doe';
```

Student Number	Student LastName	Student FirstName	Email	PhoneNumber
190	Smith	John	jsmith@usna.edu	410-431-3456
673	Doe	Jane	jdoe@usna.edu	
312	Doe	Bob	bred@usna.edu	443-451-7865

16

DELETE Statement

```
DELETE FROM table_name
[ WHERE search_condition ]
```

▪ DELETE command:

```
DELETE FROM Students
WHERE StudentNumber = 190;
```

- If you omit the WHERE clause, you will delete every row in the table!

▪ Another example:

```
DELETE FROM Departments
WHERE DepartmentName = 'ComSci'
```

Integrity constraints?!

16

The SQL SELECT Statement

▪ Basic SQL Query:

```
SELECT [DISTINCT] column_name(s) | *
FROM table_name(s)
[WHERE conditions]
```

17

Selecting All Columns: The Asterisk (*) Keyword

```
SELECT *
FROM Students;
```

Student Number	Student LastName	Student FirstName	Email	PhoneNumber	MajDeptName
190	Smith	John	jsmith@usna.edu	410-431-3456	ComSci
673	Doe	Jane	jdoe@usna.edu		ComSci
312	Doe	Jane	jdoe2@usna.edu	443-451-7865	Math

18

Specific Columns and Rows from One Table

```
SELECT      StudentNumber,
           StudentLastName,
           StudentFirstName
FROM        Students
WHERE       MajDeptName = 'ComSci';
```

Student Number	Student LastName	Student FirstName
190	Smith	John
673	Doe	Jane

19

The DISTINCT Keyword

```
SELECT SName
FROM Students;
```

StudentLastName
Smith
Doe
Doe

```
SELECT DISTINCT
SName
FROM Students;
```

StudentLastName
Smith
Doe

20

Class Exercise

- Division(Name, Building, OfficeNb)
- Department(DeptName, ChairName, WebAddress, DivName)
- Create tables
- Modify Department to add a FK constraint for DivName
- Create table Colleges with same structure as Division
- Insert everything from Division into Colleges
- Remove Division table
- Find the name of the Chair of the 'Math' Department

21

Students, Courses, Enrolled

Find the names of students enrolled in IT420

```
SELECT SName
FROM Students S, Enrolled E
WHERE S.SNb = E.SNb AND E.Cid = 'IT420'
```

SNb	SName	Email	Cid	CName	CDept
190	Smith	jsmith@usna.edu	IT420	Database	ComSci
673	Doe	jdoe@usna.edu	IT340	Networks	ComSci
312	Doe	jdoe2@usna.edu	SM121	Calculus1	Math

SNb	Cid	Semester
190	IT340	Spring2006
312	IT420	Fall2005

22

SELECT - Conceptual Evaluation Strategy

- Semantics of an SQL query defined in terms of the following conceptual evaluation strategy:
 - Compute the cross-product of table_names
 - Discard resulting rows if they fail condition
 - Delete columns that are not in column_names
 - If DISTINCT is specified, eliminate duplicate rows
- This strategy is probably the least efficient way to compute a query!
 - An optimizer will find more efficient strategies to compute the same answers.

23

Example Conceptual Evaluation

```
SELECT SName
FROM Students S, Enrolled E
WHERE S.SNb = E.SNb AND E.Cid = 'IT420'
```

S.SNb	SName	Email	E.SNb	Cid	Semester
190	Smith	jsmith@usna.edu	190	IT340	Spring2006
190	Smith	jsmith@usna.edu	312	IT420	Fall2005
673	Doe	jdoe@usna.edu	190	IT340	Spring2006
673	Doe	jdoe@usna.edu	312	IT420	Fall2005
312	Doe	jdoe2@usna.edu	190	IT340	Spring2006
312	Doe	jdoe2@usna.edu	312	IT420	Fall2005

24

Example Conceptual Evaluation

```
SELECT SName
FROM Students S, Enrolled E
WHERE S.SNb = E.SNb AND E.Cid = 'IT420'
```

S.SNb	SName	Email	E.SNb	Cid	Semester
190	Smith	jsmith@usna.edu	190	IT340	Spring2006
190	Smith	jsmith@usna.edu	312	IT420	Fall2005
673	Doe	jdoe@usna.edu	190	IT340	Spring2006
673	Doe	jdoe@usna.edu	312	IT420	Fall2005
312	Doe	jdoe2@usna.edu	190	IT340	Spring2006
312	Doe	jdoe2@usna.edu	312	IT420	Fall2005

25

Example Conceptual Evaluation

```
SELECT SName
FROM Students S, Enrolled E
WHERE S.SNb = E.SNb AND E.Cid = 'IT420'
```

SName
Doe

S.SNb	SName	Email	E.SNb	Cid	Semester
190	Smith	jsmith@usna.edu	190	IT340	Spring2006
190	Smith	jsmith@usna.edu	312	IT420	Fall2005
673	Doe	jdoe@usna.edu	190	IT340	Spring2006
673	Doe	jdoe@usna.edu	312	IT420	Fall2005
312	Doe	jdoe2@usna.edu	190	IT340	Spring2006
312	Doe	jdoe2@usna.edu	312	IT420	Fall2005

26

Modified Query

```
SELECT SNb
FROM Students S, Enrolled E
WHERE S.SNb = E.SNb AND E.Cid = 'IT420'
```

- Would the result be different with DISTINCT?

27

Sorting the Results

```
SELECT ...
FROM ...
[WHERE ...]
ORDER BY column_name(s) [ASC/DESC]
```

Example:

```
SELECT SNb, SName
FROM Students
ORDER BY SName ASC, SNb DESC
```

28

LIKE and Wildcards

```
SELECT *
FROM Students
WHERE SNb LIKE '_9_%'
```

- SQL 92 Standard (SQL Server, Oracle, etc.):
 - _ = Exactly one character
 - % = Any set of one or more characters
- MS Access
 - ? = Exactly one character
 - * = Any set of one or more characters

29

WHERE Clause Options

- AND, OR
- IN, NOT IN, BETWEEN

```
SELECT SNb
FROM Students S, Enrolled E
WHERE S.SNb = E.Nb AND
E.Cid NOT IN ('ComSci', 'Math')
```

30

Class Exercise

- Students(SNb, SName, Email)
- Courses(Cid, CName, Dept)
- Enrolled(SNb, Cid, Semester)

- Find the student number and name for each student enrolled in 'Spring2007' semester
- Find the names of all students enrolled in 'ComSci' courses

31