CS 327E Lecture 2
Shirley Cohen
January 27, 2016
Agenda

• Announcements
• Homework for today
• Reading Quiz
• Concept Questions
• Homework for next time
Announcements

• Lecture slides and notes will be posted on the course web page after each class

• Quiz and participation scores will be posted on Canvas after each class

• Please use Piazza for questions and discussion topics that are of interest to other students

• Please email or come to office hours for discussing individual questions and concerns

• Almost everyone has successfully registered their clickers! Still missing a clicker for 5 students though: Patrick Fierro, David Martinez, Kathleen Morgan, Wei-Da Pan and Luis Sanchez
Homework for Today

- Chapter 3 from the *Learning SQL* book
- Exercises at end of Chapter 3
Quiz Question 1

Which clause is mandatory in an SQL query?

A. WHERE  
B. FROM  
C. CHOOSE  
D. SELECT
Quiz Question 2

What keyword removes duplicate entries from the result set?

A. DEDUP
B. ALTER
C. UNIQUE
D. DISTINCT
Quiz Question 3

A JOIN is a mechanism for linking two tables.

A. True
B. False
Quiz Question 4

Which of the following clauses removes unwanted rows from a result set?

A. FILTER
B. HAVING
C. WHERE
D. VIEW
Quiz Question 5

Is the following query syntactically correct?

```
SELECT cust_id, fed_id, address 
FROM customer 
ORDER BY fed_id, 1;
```

A. No, since 1 does not correspond to a valid column
B. No, since the `ORDER BY` syntax requires that columns be specified in schema order
C. Yes
We have a database for a retail store that keeps information about orders in a table called `Order_Item`. How can we produce a report of all the orders that is sorted by order number?

A. `SELECT * FROM Order_Item`
B. `SELECT OrderNumber FROM Order_Item`
C. `SELECT * FROM Order_Item ORDER BY OrderNumber DESC`
D. `SELECT * FROM Order_Item ORDER BY OrderNumber`
E. None of the above
We have the same Order_Item table. This time we want to generate a report that is sorted by price from highest to lowest and then by order number.

A. SELECT *
   FROM Order_Item
   ORDER BY Price DESC, OrderNumber

B. SELECT *
   FROM Order_Item
   ORDER BY Price ASC, OrderNumber

C. SELECT *
   FROM Order_Item
   ORDER BY Price, OrderNumber

D. None of the above

E. Not enough information
Concept Question 3

Suppose we have a product catalog database as illustrated by the diagram below. How can we retrieve the model number, speed, and hard drive capacity for all the PCs that cost less than $500?

A. SELECT model, speed, hd
   FROM PC
   WHERE price < 500

B. SELECT p.model,
    pc.speed, pc.hd
   FROM Product p, PC pc
   WHERE p.model = pc.model
   AND price < 500

C. All of the above

D. None of the above

E. Not enough information
Concept Question 4

How can we find the model number, speed, and hard drive capacity of all PCs that have a 12x or 24x CD drive and that cost less than $600?

A. SELECT model, speed, hd
   FROM PC
   WHERE price < 600
   AND cd = '12x'
   OR cd = '24x'

B. SELECT model, speed, hd
   FROM PC
   WHERE price < 600
   AND cd IN ('12x', '24x')

C. SELECT model, speed, hd
   FROM PC
   WHERE price < 600
   AND cd BETWEEN '12x' AND '24x'

D. None of the above
We are building a database that tracks projects worked on by software development teams. Each project has one or more developers and they can be a lead, senior or junior developer on the project. Based on these requirements, suggest how to add some integrity checking to the Teams table.

A. CONSTRAINT project_id_fk
FOREIGN KEY project_id
REFERENCES Projects(project_id)

B. CONSTRAINT emp_id_fk
FOREIGN KEY emp_id
REFERENCES Employees(emp_id)

C. CONSTRAINT role_ck CHECK(role
IN ('L', 'S', 'J'))

D. CONSTRAINT proj_emp_pk
PRIMARY KEY (project_id, emp_id)

E. All of the above

create table Projects
(
    project_id INTEGER PRIMARY KEY,
    start_date DATE NOT NULL,
    ...
)

create table Employees
(
    emp_id INTEGER PRIMARY KEY,
    first_name VARCHAR(20),
    last_name VARCHAR(20),
    ...
)

create table Teams
(
    project_id INTEGER NOT NULL,
    emp_id INTEGER NOT NULL,
    role CHAR(1),
    ...
)
Homework for Next Time

- Chapter 4 from the Learning SQL book