

CS327E Lecture 2 - Wednesday 1/27/2016

Reading Quiz:

Question 1: Ans = D

Question 2: Ans = D

Question 3: Ans = A

Question 4: Ans = C

Question 5: Ans = C

Concept Questions:

1. 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;

This is incorrect because it does not handle the ordering.

b. **SELECT OrderNumber**

FROM Order_Item;

This is incorrect because the result set is OrderNumber and it also does not handle the sorting.

c. **SELECT ***

FROM Order_Item

ORDER BY OrderNumber DESC;

This is incorrect because the result set is ordered in descending order.

d. **SELECT ***

FROM Order_Item

ORDER BY OrderNumber;

This is the correct answer because the question is asking for ASC order.

e. None of the above

2. 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

This is the correct answer because it sorts by price in descending order and then the order number.

- b. `SELECT *`
`FROM Order_Item`
`ORDER BY Price ASC, OrderNumber`
This is not the correct answer because Price is sorted by ascending order.
 - c. `SELECT *`
`FROM Order_Item`
`ORDER BY Price, OrderNumber`
This is not the correct answer because Price is sorted by ascending order.
 - d. None of the above
 - e. Not enough information
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`
This is the correct answer, because it grabs all the data that is needed without doing an unnecessary join.
 - b. `SELECT p.model, pc.speed, pc.hd`
`FROM Product p, PC pc`
`WHERE p.model = pc.model`
`AND price < 500`
Although this could work, the join is unnecessary.
 - c. All of the above
 - d. None of the above
 - e. Not enough information

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'`

This is the incorrect answer because it will include CDs that are '24x' and have prices greater than or equal to 600. A way to resolve this is to use parentheses such that it looks like `WHERE price < 600 AND (cd = '12x' OR cd = '24x')`.

b. **`SELECT model, speed, hd
FROM pc
WHERE price < 600
AND cd IN ('12x', '24x')`**

This is the correct answer because 'IN' is similar to an 'OR' statement in this case. It will check whether or not cd is equal to '12x' or if it is equal to '24x'.

c. `SELECT model, speed, hd FROM pc
WHERE price < 600
AND cd BETWEEN '12x' AND '24x'`

The keyword 'BETWEEN' is usually for an inclusive range. If a cd was equal to '15x', then it would be included in the answer.

d. None of the above

5. 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)`

This constraint will ensure that project_id does exist, thus preserving data integrity.

b. `CONSTRAINT emp_id_fk
FOREIGN KEY emp_id
REFERENCES Employees(emp_id)`

Similar to the project_id, we want to make sure that the emp_id value exists, before adding it in.

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

We are limiting the roles to either null, 'L', 'S', or 'J' to represent lead, senior, junior, or an employee without a title.

- d. CONSTRAINT proj_emp_pk
PRIMARY KEY (project_id, emp_id)

This is a primary key constraint, which is a composite key of the project_id and emp_id. We also want to make sure that we don't have an employee that works on the same project in two different roles.

- e. **All of the above**