CS 327E Lecture 8

Shirley Cohen

October 19, 2016

Homework for Today

- Chapters 3 and 4 from the Learning SQL book
- Exercises at the end of assigned chapters

Quiz Question 1

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

A. `FILTER`
B. `GROUP BY`
C. `WHERE`
D. `VIEW`

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 one of the following will return only the NULL records?

```
A. SELECT * FROM TableName
   WHERE ColumnName IS NULL

B. SELECT * FROM TableName
   WHERE ColumnName == NULL

C. SELECT * FROM TableName
   WHERE ColumnName = NULL

D. SELECT * FROM TableName
   WHERE ColumnName = 'NULL'
```

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

```
mysql> describe customer;
+--------------+------------------+
| Field        | Type             |
+--------------+------------------+
| cust_id      | int(10) unsigned |
| fed_id       | varchar(12)      |
| cust_type_cd | enum('I','B')    |
| address      | varchar(30)      |
| city         | varchar(20)      |
| state        | varchar(20)      |
| postal_code  | varchar(10)      |
+--------------+------------------+
```

Concept Question 1

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?

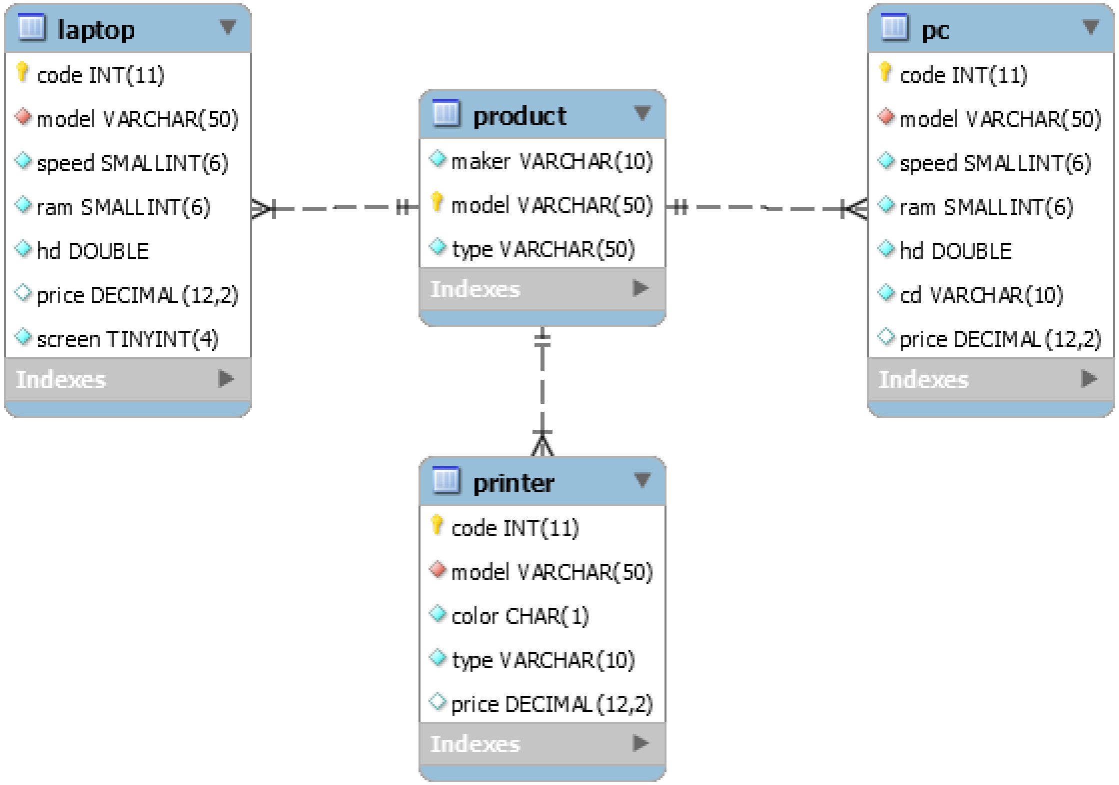
A.
```
SELECT model, speed, hd
FROM PC
WHERE price < 500
```

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

C. Either A or B

D. Neither A or B

E. Not enough information



Concept Question 2

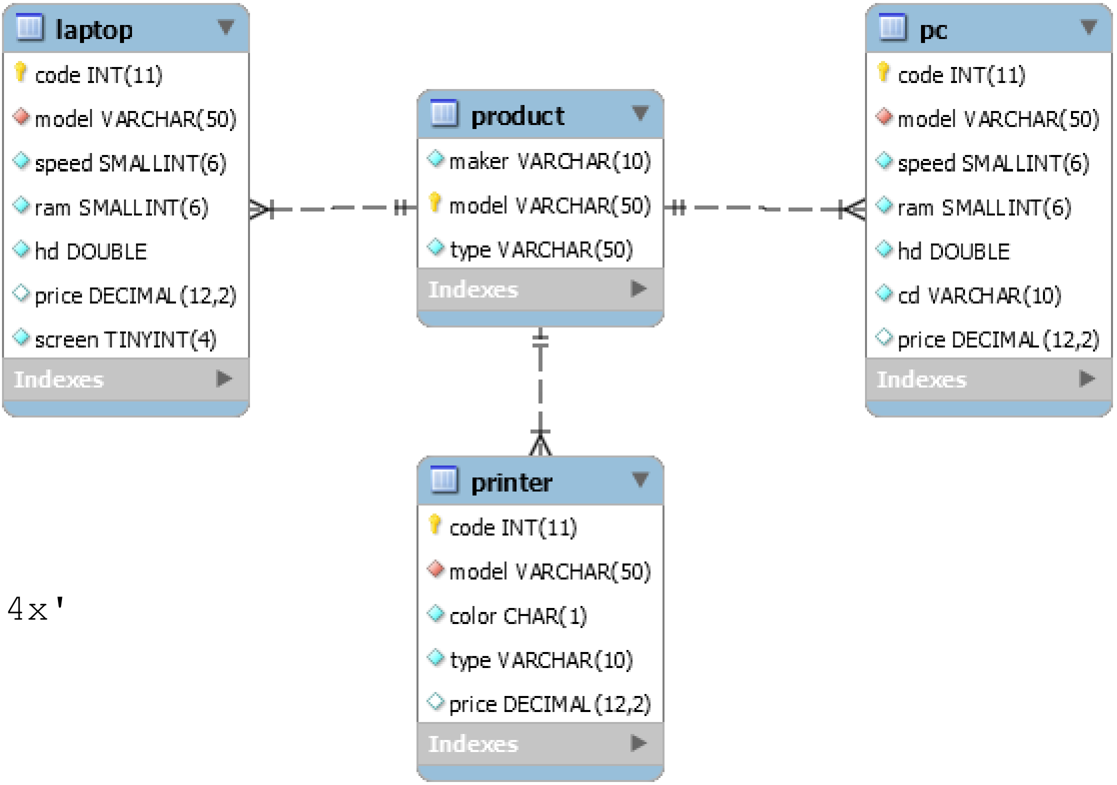
How can we find the model number, speed, and hard drive capacity of all PCs that have a 12x or 24x CD drives and that cost less than $600?

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 = '12x' AND cd = '24x'
```

D. None of the above



Concept Question 3

Suppose we have a database for a retail store. This database tracks customers, orders, inventory, products, and vendors. We want to obtain a list of our vendors, but we are only interested in those vendors who are **not** based in Austin. What SQL query can we use to retrieve all vendors who are not local?

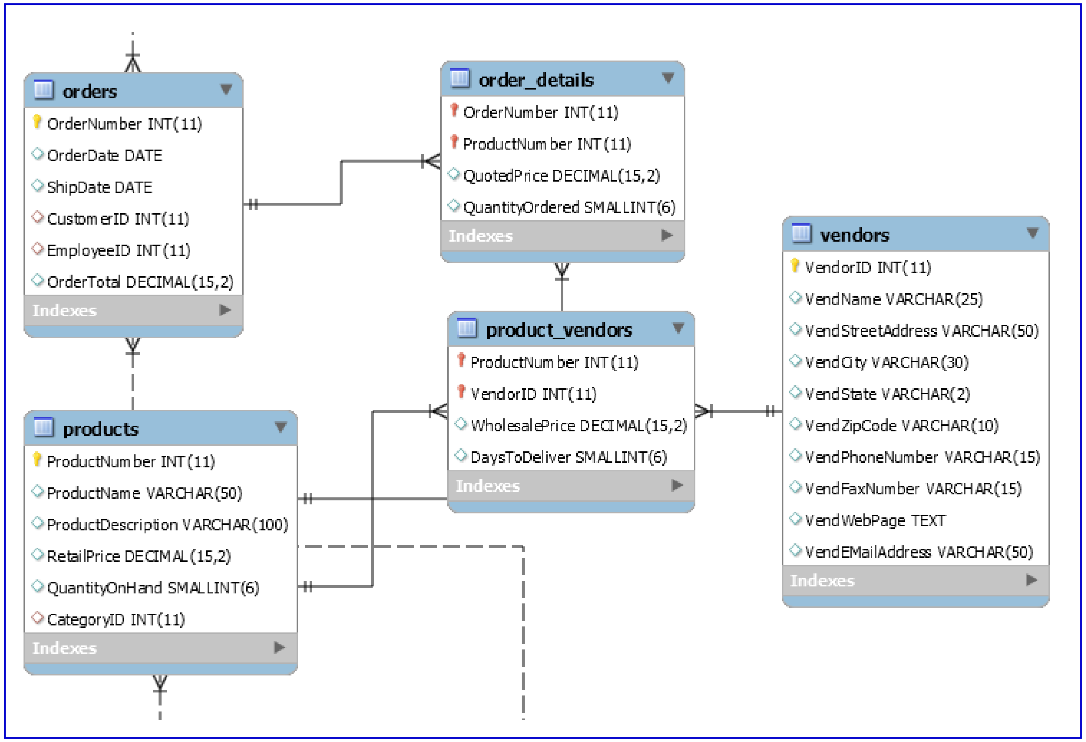
A.
```
select vendName
from Vendors
where vendCity <>
'AUSTIN'
```

B.
```
select vendName
from Vendors
where
UPPER(vendCity) <>
'AUSTIN'
```

C.
```
select vendName
from Vendors
where UPPER(vendCity)
<> 'AUSTIN' or
vendCity is null
```

D. Any of the above

E. None of the above



orders
- OrderNumber INT(11)
- OrderDate DATE
- ShipDate DATE
- CustomerID INT(11)
- EmployeeID INT(11)
- OrderTotal DECIMAL(15,2)
- Indexes

order_details
- OrderNumber INT(11)
- ProductNumber INT(11)
- QuotedPrice DECIMAL(15,2)
- QuantityOrdered SMALLINT(6)
- Indexes

vendors
- VendorID INT(11)
- VendName VARCHAR(25)
- VendStreetAddress VARCHAR(50)
- VendCity VARCHAR(30)
- VendState VARCHAR(2)
- VendZipCode VARCHAR(10)
- VendPhoneNumber VARCHAR(15)
- VendFaxNumber VARCHAR(15)
- VendWebPage TEXT
- VendEMailAddress VARCHAR(50)
- Indexes

product_vendors
- ProductNumber INT(11)
- VendorID INT(11)
- WholesalePrice DECIMAL(15,2)
- DaysToDeliver SMALLINT(6)
- Indexes

products
- ProductNumber INT(11)
- ProductName VARCHAR(50)
- ProductDescription VARCHAR(100)
- RetailPrice DECIMAL(15,2)
- QuantityOnHand SMALLINT(6)
- CategoryID INT(11)
- Indexes

Concept Question 4

We have a retail store database that keeps information about the items belonging to an order in a table `Order_Details`. How can we produce a report of all the order items that is first sorted by `OrderNumber` (lowest to highest) and secondly sorted by `Price` (highest to lowest)?

A.
```
SELECT *
FROM Order_Details
```

B.
```
SELECT OrderNumber
FROM Order_Details
ORDER BY
OrderNumber, Price
```

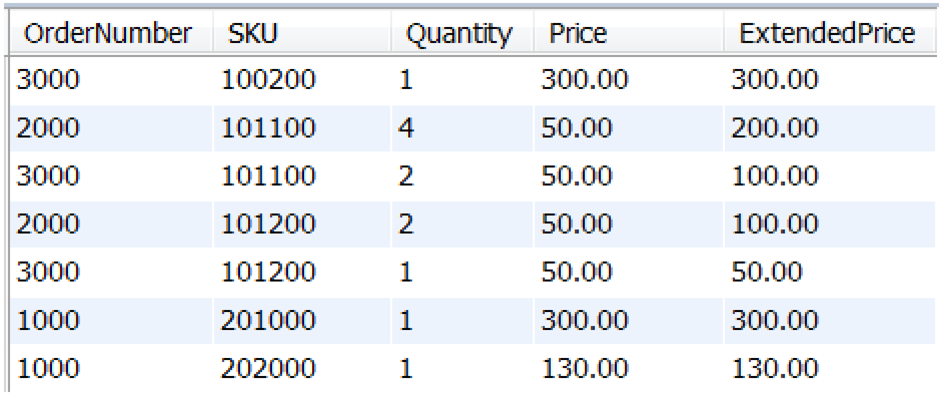
C.
```
SELECT *
FROM Order_Details
ORDER BY
OrderNumber DESC,
Price DESC
```

D.
```
SELECT *
FROM Order_Details
ORDER BY
OrderNumber ASC,
Price DESC
```

E. None of the above

Order_Details (<u>OrderNumber</u>, <u>SKU</u>, Quantity, Price, ExtendedPrice)

SELECT * FROM Order_Details



OrderNumber	SKU	Quantity	Price	ExtendedPrice
3000	100200	1	300.00	300.00
2000	101100	4	50.00	200.00
3000	101100	2	50.00	100.00
2000	101200	2	50.00	100.00
3000	101200	1	50.00	50.00
1000	201000	1	300.00	300.00
1000	202000	1	130.00	130.00

Concept Question 5

We have the same `Order_Details` table as before. This time we want to generate a report of all the items or SKUs that have ever been ordered. We only want a single record per SKU regardless of how many orders it belongs to. We also want to sort the results by SKU from lowest to highest.

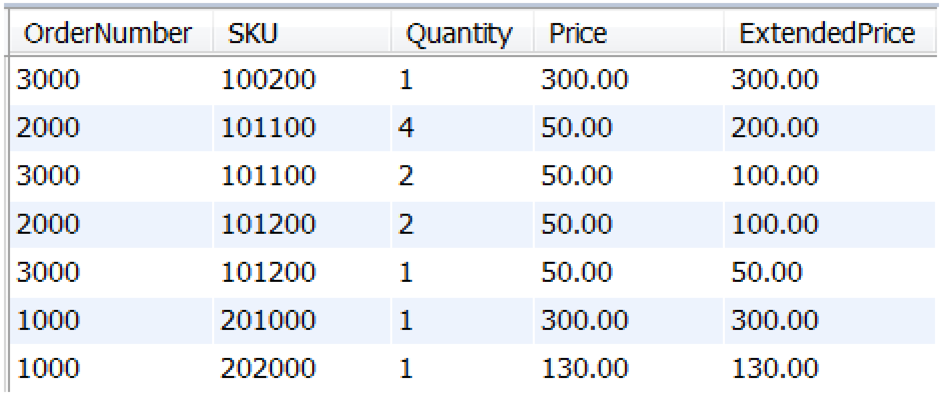
```
A.    SELECT *
      FROM Order_Details
      ORDER BY SKU

B.    SELECT DISTINCT SKU
      FROM Order_Details
      ORDER BY SKU DESC

C.    SELECT DISTINCT SKU
      FROM Order_Details
      ORDER BY SKU ASC
```

D. None of the above

E. Not enough information

Order_Details (OrderNumber, SKU, Quantity, Price, ExtendedPrice)

SELECT * FROM Order_Details



OrderNumber	SKU	Quantity	Price	ExtendedPrice
3000	100200	1	300.00	300.00
2000	101100	4	50.00	200.00
3000	101100	2	50.00	100.00
2000	101200	2	50.00	100.00
3000	101200	1	50.00	50.00
1000	201000	1	300.00	300.00
1000	202000	1	130.00	130.00

Concept Question 6

We have a database that tracks software defects. We want to look-up all the defects that are both **unassigned** and **active**. Assume that the `assigned_to` field indicates that a defect has been assigned when it is not null. Assume that an **active** defect means a `status` of not `'CLOSED'`.

A.

```
select * from Defects
where assigned_to IS NULL
and (status <> 'CLOSED'
or status IS NULL)
```

B.

```
select * from Defects
where assigned_to IS NULL
and status <> 'CLOSED'
```

C.

```
select * from Defects
where assigned_to = NULL
and (status <> 'CLOSED'
or status = NULL)
```

D.

```
select * from Defects
where assigned_to IS NULL
and status NOT IN
('CLOSED')
```

E. None of the above

```
CREATE TABLE Accounts (
 account_id INT PRIMARY KEY,
 account_name VARCHAR(20),
 first_name VARCHAR(20),
 last_name VARCHAR(20),
 email VARCHAR(100),
 password_hash CHAR(64),
 ...);

CREATE TABLE Defects (
 bug_id INT PRIMARY KEY,
 date_reported DATE NOT NULL,
 summary VARCHAR(80),
 reported_by INT NOT NULL,
 assigned_to INT,
 status enum('NEW', 'OPEN', 'QA', 'CLOSED'),
 ...
 FOREIGN KEY (reported_by) REFERENCES
Accounts(account_id),
 FOREIGN KEY (assigned_to) REFERENCES
Accounts(account_id));
```

Homework

- Read chapters 5 and 10 from the Learning SQL book
- Exercises at the end of assigned chapters