

0/30 Questions Answered
30 questions with unsaved changes

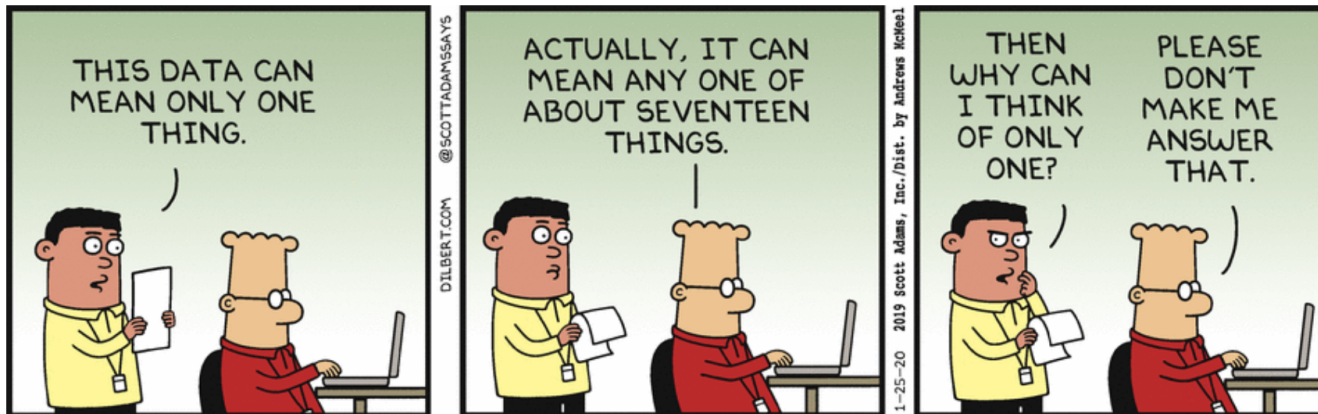
Midterm 2

STUDENT NAME

Search students by name or email...

Q1 Preliminaries

1 Point



Instructions (1 point!)

The allowed time for the exam is 60 minutes (50 minutes plus a 10 minute technology buffer).
Be sure to pay attention to time and budget your time accordingly!

The exam is open pre-prepared cheat sheet, open book, open notes, open web browser, and even open MySQL. You are just not allowed to communicate with or otherwise interact with other students (or friends) during the course of the exam, and this includes your HW brainstorming buddy. This exam is to be a solo effort!

Read each question carefully, in its entirety, and then answer each part of the question. If you don't understand something, please just make your best educated guess and proceed accordingly.

Acknowledgement: I certify that I am taking this exam myself, on my own, with honesty and integrity, without interaction with others during the exam, and without having obtained any information about the exam's content from others prior to taking it.

☒ True

☐ False

Reference Data

The exam questions that involve query-writing and data will all be based on the following inventory management schema about suppliers, parts, and which suppliers supply which parts. (You may wish to scroll back periodically to answer some of the questions. Alternatively, you may want to take a screen shot to have in a separate browser window for viewing.)

Note: You will not need to copy/paste anything from the schema or data below, because later, when you might want a subset of this data in CSV form, we will give you what you'll need in a copy/paste-able form there.

```
CREATE TABLE Supplier (  
  sno INT NOT NULL,  
  sname VARCHAR(40),  
  email VARCHAR(60),  
  state CHAR(2).
```

```

-- Create Supplier table
CREATE TABLE Supplier (
    sno INT NOT NULL,
    sname VARCHAR(40) NULL,
    email VARCHAR(100) NULL,
    city VARCHAR(20) NULL,
    state VARCHAR(2) NULL,
    zip INT NULL,
    since DATE NULL,
    PRIMARY KEY (sno)
);

-- Create Part table
CREATE TABLE Part (
    pno INT NOT NULL,
    pname VARCHAR(40) NULL,
    color VARCHAR(60) NULL,
    price DECIMAL(8,2),
    PRIMARY KEY (pno)
);

-- Create Supply table
CREATE TABLE Supply (
    sno INT NOT NULL,
    pno INT NOT NULL,
    quantity INT,
    discount DECIMAL(4,2),
    since DATE NULL,
    PRIMARY KEY (sno, pno)
);

-- Insert Supplier values
INSERT INTO Supplier VALUES
(1, 'Acme', 'sales@acme.com', 'NY', 30, '2020-11-01'),
(2, 'BuyNow', 'deals@buynow.com', 'CA', 150, '2020-11-21'),
(3, 'WorstBuy', 'mgr@worstbuy.com', 'IL', 100, '2020-10-31'),
(4, 'WorstBuy', 'mgr@worstbuy.com', 'CA', 0, '2020-10-31');

-- Insert Part values
INSERT INTO Part VALUES
(1, 'Widget', 'blue', 10.00),
(2, 'Widget', 'teal', 20.00),
(3, 'Gadget', 'purple', 20.0),
(4, 'Fidget', 'blue', NULL);

-- Insert Supply values
INSERT INTO Supply VALUES
(1, 1, 20, 0.20, '2020-11-01'),
(1, 3, 10, 0.20, '2020-11-03'),
(2, 1, 40, 0.10, '2020-11-21'),
(2, 2, 60, 0.30, '2020-11-22'),
(3, 3, 50, 0.20, '2020-11-23')

```

```
(2, 3, 50, 0.20, '2020-11-25'),  
(3, 3, 100, 0.00, '2020-10-31');
```

[Save Answer](#)***Unsaved Changes**

Q2 Short and Two the Point(s)

26 Points

For each of the following questions, you should

```
SELECT best(answer) FROM question LIMIT 1.
```

Q2.1

2 Points

Division is a core (necessary) operation in the relational algebra.

☐ True☒ False[Save Answer](#)***Unsaved Changes**

Q2.2

2 Points

Cross product is a core (necessary) operation in the relational algebra.

☒ True

☐ False

Save Answer

***Unsaved Changes**

Q2.3

2 Points

The Python programming language is relationally complete.

☒ True

☐ False

Save Answer

***Unsaved Changes**

Q2.4

2 Points

It is possible to write a relational algebra query that produces a result with infinitely many tuples.

☐ True

☒ False

Save Answer

***Unsaved Changes**

Q2.5

2 Points

Based on the given inventory tables, what is the cardinality of the result of the relational algebra query: $\pi_{email} (Supplier \times Part)$

☐ 0☒ 3☐ 4☐ 12☐ 16☐ ∞ Save Answer***Unsaved Changes****Q2.6**

2 Points

Based on the given inventory tables, what is the cardinality of the result of the relational calculus query: $\{ s \mid (s \in \text{Supplier}) \wedge (s.\text{state} = \text{'CA'}) \wedge (s \notin \text{Supply}) \}$

- ☐ 0
- ☐ 1
- ☒ 2
- ☐ 3
- ☐ 4
- ☐ ∞

Save Answer

Unsaved Changes*Q2.7**

2 Points

What would be the most appropriate MySQL mechanism to employ to ensure that the part count (i.e., numparts) of a Supplier is always correct (i.e., consistent with the data in the Supply table)?

- ☐ FOREIGN KEY CONSTRAINT
- ☐ CHECK CONSTRAINT
- ☐ UNIQUE CONSTRAINT
- ☒ TRIGGER
- ☐ VIEW

Save Answer

***Unsaved Changes**

Q2.8

2 Points

What would be the most appropriate MySQL mechanism to employ to ensure that the email address of a supplier is always in the .com domain?

- ☐ FOREIGN KEY CONSTRAINT
- ☒ CHECK CONSTRAINT
- ☐ UNIQUE CONSTRAINT
- ☐ TRIGGER
- ☐ VIEW

Save Answer***Unsaved Changes****Q2.9**

2 Points

What would be the most appropriate MySQL mechanism to employ to ensure that no two suppliers have the same email address?

- ☐ FOREIGN KEY CONSTRAINT
- ☐ CHECK CONSTRAINT
- ☒ UNIQUE CONSTRAINT
- ☐ TRIGGER
- ☐ VIEW

Save Answer***Unsaved Changes**

Q2.10

2 Points

What would be the most appropriate MySQL mechanism to employ to ensure that each of the suppliers being referred to by a supply entry exists?

☒ FOREIGN KEY CONSTRAINT☐ CHECK CONSTRAINT☐ UNIQUE CONSTRAINT☐ TRIGGER☐ VIEWSave Answer***Unsaved Changes****Q2.11**

2 Points

How many rows will the following MySQL query return given the given inventory tables?

```
SELECT p.pname, p.price * (1.00 - s.discount) AS dprice
FROM Part p JOIN Supply s ON p.pno = s.pno
WHERE p.pname = 'Fidget' AND p.price != 100.00;
```

☒ 0☐ 1☐ 2☐ 4Save Answer***Unsaved Changes****Q2.12**

2 Points

What avgprice value will the following MySQL query return based on the given inventory tables?

```
SELECT SUM(p.price) AS avgprice  
FROM Part p  
WHERE p.color = 'blue';
```

☐ NULL☐ 0.00☒ 10.00☐ 130.00Save Answer***Unsaved Changes**

Q2.13

2 Points

Is the following MySQL query legal according to the SQL standard?

```
SELECT p.color, AVG(p.price) AS cnt
FROM Part p
WHERE p.pname LIKE '%get'
GROUP BY p.color
HAVING COUNT(DISTINCT p.pname) > 1;
```

☒ Yes☐ No[Save Answer](#)***Unsaved Changes****Q3** Taking an Equivalence Class

30 Points

Consider once again the tables of our inventory database:

```
CREATE TABLE Supplier (
  sno INT NOT NULL,
  sname VARCHAR(40),
  email VARCHAR(60),
  state CHAR(2),
  numparts INT,
  since DATE,
  PRIMARY KEY (sno)
);
```

```
CREATE TABLE Part (  
    pno INT NOT NULL,  
    pname VARCHAR(40) NULL,  
    color VARCHAR(60) NULL,  
    price DECIMAL(8,2),  
    PRIMARY KEY (pno)  
);  
  
CREATE TABLE Supply (  
    sno INT NOT NULL,  
    pno INT NOT NULL,  
    quantity INT,  
    discount DECIMAL(4,2),  
    since DATE NULL,  
    PRIMARY KEY (sno, pno)  
);  
  
INSERT INTO Supplier VALUES  
    (1, 'Acme', 'sales@acme.com', 'NY', 30, '2020-11-01'),  
    (2, 'BuyNow', 'deals@buynow.com', 'CA', 150, '2020-11-21'),  
    (3, 'WorstBuy', 'mgr@worstbuy.com', 'IL', 100, '2020-10-31'),  
    (4, 'WorstBuy', 'mgr@worstbuy.com', 'CA', 0, '2020-10-31');  
  
INSERT INTO Part VALUES  
    (1, 'Widget', 'blue', 10.00),  
    (2, 'Widget', 'teal', 20.00),  
    (3, 'Gadget', 'purple', 20.0),  
    (4, 'Fidget', 'blue', NULL);  
  
INSERT INTO Supply VALUES  
    (1, 1, 20, 0.20, '2020-11-01'),  
    (1, 3, 10, 0.20, '2020-11-03'),  
    (2, 1, 40, 0.10, '2020-11-21'),  
    (2, 2, 60, 0.30, '2020-11-22'),  
    (2, 3, 50, 0.20, '2020-11-23'),  
    (3, 3, 100, 0.00, '2020-10-31');
```

For each of the following problems, you will be given a relational query or an English query along with a set of additional queries. For each problem, indicate which query or queries in the additional query set are equivalent to the given query. (Note that equivalent means that it will produce the same result as the original query for *all possible database states* permitted by the HR schema's CREATE TABLE statements.) **Note:** Be sure to consider each query *carefully*!

Q3.1

8 Points

Here is the given query:

```
 $\pi$  sname (Supplier  $\bowtie$  (sno = sno) Supply)
```

Select the equivalent query or queries (if any):

☐ SELECT sname FROM Supplier s, Supply y WHERE s.sno = y.sno

☐ π sname (Supplier \bowtie Supply)

☒ SELECT sname FROM Supplier s WHERE EXISTS (SELECT * FROM Supply WHERE sno = s.sno)

☐ SELECT sname FROM Supplier s JOIN Supply y ON s.sno = y.sno

Save Answer

***Unsaved Changes**

Q3.2

4 Points

Here is the given query:

```
SELECT s.sno
FROM Supplier s
WHERE
  (SELECT COUNT(DISTINCT p.pno)
   FROM Part p, Supply y
   WHERE pname = 'Widget'
        AND p.pno = y.pno AND y.sno = s.sno)
= (SELECT COUNT(pno)
   FROM Part p
   WHERE pname = 'Widget')
```

Select the equivalent query or queries (if any):

☒ $\{ t(sno) \mid \forall p \in \text{Part } (p.pname = \text{'Widget'} \Rightarrow (\exists y \in \text{Supply } (y.pno = p.pno \wedge t.sno = s.sno))) \}$ ☐ $\text{Supply} \div (\sigma \text{ pname} = \text{'Widget'} (\text{Part}))$ [Save Answer](#)***Unsaved Changes****Q3.3**

6 Points

Here is the given query:

```
SELECT pname FROM Part WHERE price IS NULL
```

Select the equivalent query or queries (if any):

☐ SELECT pname FROM Part WHERE pno = 4

☒ SELECT pname FROM Part WHERE pno NOT IN (SELECT pno FROM Part WHERE price <= 0.00 OR price >= 0.00)

☐ SELECT DISTINCT pname FROM Part WHERE price IS NULL

Save Answer

***Unsaved Changes**

Q3.4

6 Points

Here is the given query:

```
SELECT s.sno, s.sname, COUNT(y.pno)
FROM Supplier s, Supply y
WHERE s.sno = y.sno
GROUP BY s.sno, s.sname
```

Select the equivalent query or queries (if any):

☐ SELECT s.sno, s.sname, COUNT(y.pno) FROM Supplier s LEFT JOIN Supply y ON s.sno = y.sno GROUP BY s.sno, s.sname

✓ SELECT s.sno, s.sname, COUNT(y.pno) FROM Supplier s INNER JOIN Supply y ON s.sno = y.sno GROUP BY s.sno, s.sname

✓ SELECT s.sno, s.sname, COUNT(y.pno) FROM Supplier s, Supply y WHERE s.sno = y.sno GROUP BY s.sno

Save Answer

***Unsaved Changes**

Q3.5

6 Points

Here is the given query:

$(\pi \text{ sname } (\sigma \text{ state} = \text{'CA'} \wedge \text{numparts} \leq 100 (\text{Supplier})))$

Select the equivalent query or queries (if any):

☐ $(\pi \text{ sname } (\sigma \text{ state} = \text{'CA'} (\text{Supplier}))) \cap (\pi \text{ sname } (\sigma \text{ numparts} \leq 100 (\text{Supplier})))$

✓ $\pi \text{ sname } (\text{Supplier} \bowtie ((\pi \text{ sno } (\sigma \text{ state} = \text{'CA'} (\text{Supplier}))) \cap (\pi \text{ sno } (\sigma \text{ numparts} \leq 100 (\text{Supplier}))))$

☐ $(\pi \text{ sno } (\sigma \text{ state} = \text{'CA'} (\text{Supplier}))) \cap (\pi \text{ sno } (\sigma \text{ numparts} \leq 100 (\text{Supplier})))$

[Save Answer](#)***Unsaved Changes**

Q4 Query Me This, Batman

20 Points

It's time to check the box that says "I'm tired of checking boxes!" In this problem you will actually write or "hand execute" a couple of queries of your **own!** Exciting, right?

Note: When asked to show queries' answers below, feel free to show them in a CSV-like format. For example, suppose the given query is:

```
SELECT * FROM Part WHERE pno > 2;
```

In this case you could denote the query's answer as follows (since Gradescope won't let you hand-draw pictures of tables with rows and columns).

```
pno, pname, color, price
3, Gadget, purple, 20.00
4, Fidget, blue, NULL
```

Note: Be sure to include the first row, listing the result column names, as well as the data rows!

In preparation for what follows, here's a repeat of the inventory database's tables and their current contents:

```
CREATE TABLE Supplier (  
  sno INT NOT NULL,  
  sname VARCHAR(40),  
  email VARCHAR(60),  
  state CHAR(2).
```

```

    numparts INT,
    since DATE,
    PRIMARY KEY (sno)
);

CREATE TABLE Part (
    pno INT NOT NULL,
    pname VARCHAR(40) NULL,
    color VARCHAR(60) NULL,
    price DECIMAL(8,2),
    PRIMARY KEY (pno)
);

CREATE TABLE Supply (
    sno INT NOT NULL,
    pno INT NOT NULL,
    quantity INT,
    discount DECIMAL(4,2),
    since DATE NULL,
    PRIMARY KEY (sno, pno)
);

INSERT INTO Supplier VALUES
    (1, 'Acme', 'sales@acme.com', 'NY', 30, '2020-11-01'),
    (2, 'BuyNow', 'deals@buynow.com', 'CA', 150, '2020-11-21'),
    (3, 'WorstBuy', 'mgr@worstbuy.com', 'IL', 100, '2020-10-31'),
    (4, 'WorstBuy', 'mgr@worstbuy.com', 'CA', 0, '2020-10-31');

INSERT INTO Part VALUES
    (1, 'Widget', 'blue', 10.00),
    (2, 'Widget', 'teal', 20.00),
    (3, 'Gadget', 'purple', 20.0),
    (4, 'Fidget', 'blue', NULL);

INSERT INTO Supply VALUES
    (1, 1, 20, 0.20, '2020-11-01'),
    (1, 3, 10, 0.20, '2020-11-03'),
    (2, 1, 40, 0.10, '2020-11-21'),
    (2, 2, 60, 0.30, '2020-11-22'),
    (2, 3, 50, 0.20, '2020-11-23')

```

```
(2, 5, 50, 0.20, '2020-11-25'),  
(3, 3, 100, 0.00, '2020-10-31');
```

Q4.1

8 Points

Write a SQL query to find the part numbers, part names, and prices of parts that are blue or teal. Order your query results from most to least expensive.

```
SELECT pno, pname, price  
FROM Part  
WHERE color = 'blue'  
       OR color = 'teal'  
ORDER BY price DESC;
```

Save Answer

***Unsaved Changes**

Q4.2

9 Points

Consider the following view definition SV:

```
CREATE VIEW SV AS
SELECT sname,
       COUNT(DISTINCT email) AS emails,
       COUNT(DISTINCT state) AS states,
       SUM(numparts) AS totparts
FROM Supplier
GROUP BY sname;
```

What will the following query over SV return?

```
SELECT * FROM SV ORDER BY totparts DESC;
```

```
sname,emails,states,totparts
BuyNow,1,1,150
WorstBuy,1,2,100
Acme,1,1,30
```

Save Answer

***Unsaved Changes**

Q4.3

3 Points

As you know, views are virtual tables, and some views are updatable. Would the following update be possible on this view?

```
UPDATE SV
```

```
SET totparts = totparts + 50
```

WHERE sname = 'WorstBuy';

☐ Yes

☒ No

Save Answer

***Unsaved Changes**

Q5 To Err is Human, to Succeed Divine

23 Points

It's time to move past SELECT queries and views and put some of SQL's integrity features as well as its more advanced language features to use on the inventory database.

Q5.1

2 Points

One of the inventory application developers just added the following trigger to enforce one of the business rules for this inventory management system:

```
DELIMITER //
```



```
CREATE TRIGGER CheckIt
BEFORE INSERT ON Supplier
FOR EACH ROW
BEGIN
    IF EXISTS (SELECT * FROM Supplier s
               WHERE s.sname = NEW.sname
                  AND s.state = NEW.state)
    THEN
        SIGNAL SQLSTATE '03000'
        SET MESSAGE_TEXT = 'Dude, really?';
    END IF;
END;
```

Briefly state (in English) the business rule that this trigger is designed to enforce.

The trigger enforces a rule that no two suppliers can have the same name in the same state.

Now show what the system's response will be to each of the following SQL statements. To get you started, we'll do one for you:

Statement:

```
SELECT * FROM Supplier;
```

Response:

sno, sname, email, state, numparts, since

1, Acme, sales@acme.com, NY, 30, 2020-11-01

2, BuyNow, deals@buynow.com, CA, 150, 2020-11-21

3, WorstBuy, mgr@worstbuy.com, IL, 100, 2020-10-31

4, WorstBuy, mgr@worstbuy.com, CA, 0, 2020-10-31

(Note: Feel free to copy/paste from the above data if you need or want it later on to save yourself some typing.)

For each INSERT statement, indicate whether it succeeds or fails. In the event of an error, use the text box to indicate what the nature of the error message would be. As you answer each of the questions, be sure to look carefully at the table definitions as well as the state of the database and the incoming row's field values. (This problem is about more than just the trigger!)

Save Answer

***Unsaved Changes**

Q5.2

2 Points

Statement:

```
INSERT INTO Supplier
VALUES (5, 'AAAA', 'mgr@aaaa.com', 'NY', 0, '2020-11-23');
```

Response:

☐ Error.

☒ Success.

Enter your answer here

Save Answer

***Unsaved Changes**

Q5.3

2 Points

Statement:

```
INSERT INTO Supplier  
VALUES (1, 'Acme', 'sales@acme.com', 'CO', 0, '2020-11-23');
```

Response:

☒ Error.

☐ Success.

Error: Duplicate primary key.

Save Answer

***Unsaved Changes**

Q5.4

2 Points

Statement:

```
INSERT INTO Supplier
VALUES ('six', 'PriceCo', 'sales@priceco.com', 'OR', 0, '2020-11-23');
```

Response:☒ Error.☐ Success.

Error: Wrong data type for column sno.

[Save Answer](#)***Unsaved Changes****Q5.5**

2 Points

Statement:

```
INSERT INTO Supplier
VALUES (7, 'BuyNow', 'deals@buynow.com', 'MA', 0, '2020-11-23');
```

Response:

☐ Error.

☒ Success.

Enter your answer here

Save Answer

***Unsaved Changes**

Q5.6

2 Points

Statement:

```
INSERT INTO Supplier  
VALUES (8, 'BuyNow', 'deals@buynow.com', 'MA', 0, '2020-11-23');
```

Response:

☒ Error.

☐ Success.

Error: Dude, really?

Save Answer

***Unsaved Changes**

Q5.7

8 Points

Statement:

```
SELECT * FROM Supplier ORDER BY sname ASC;
```

Response:

```
sno, sname, email, state, numparts, since  
5, AAAA, mgr@aaaa.com, NY, 0, 2020-11-23  
1, Acme, sales@acme.com, NY, 30, 2020-11-01  
2, BuyNow, deals@buynow.com, CA, 150, 2020-11-21  
7, BuyNow, deals@buynow.com, MA, 0, 2020-11-23  
3, WorstBuy, mgr@worstbuy.com, IL, 100, 2020-10-31  
4, WorstBuy, mgr@worstbuy.com, CA, 0, 2020-10-31
```

[Save Answer](#)***Unsaved Changes****Q5.8**

3 Points

Last but not least, let's consider SQL's access control features. Consider the following sequence of events:

Sue: CREATE VIEW V AS ...;

Sue: GRANT SELECT ON V TO Tom WITH GRANT OPTION;

Tom: GRANT SELECT ON V TO Rose;

Which of the following (if any) can Tom do now (assuming that V is an updatable view)?

☒ SELECT * ON V;

☒ GRANT SELECT ON V to Abdul;

☐ INSERT INTO V VALUES (...);

Save Answer

***Unsaved Changes**

Save All Answers

Submit & View Submission >