Information Management II
Midterm Examination

Instructions: There are two sets of questions in this exam. Make sure to answer the question in a sequential order as they are related. Each SET OF QUESTIONS are independent of each other.

This is not an open book nor reference exam. You are not allowed to open any other applications nor websites aside from the following:

- WORD
- DIAGRAMING TOOL
- CANVAS
- PHPMYADMIN/XAMPP

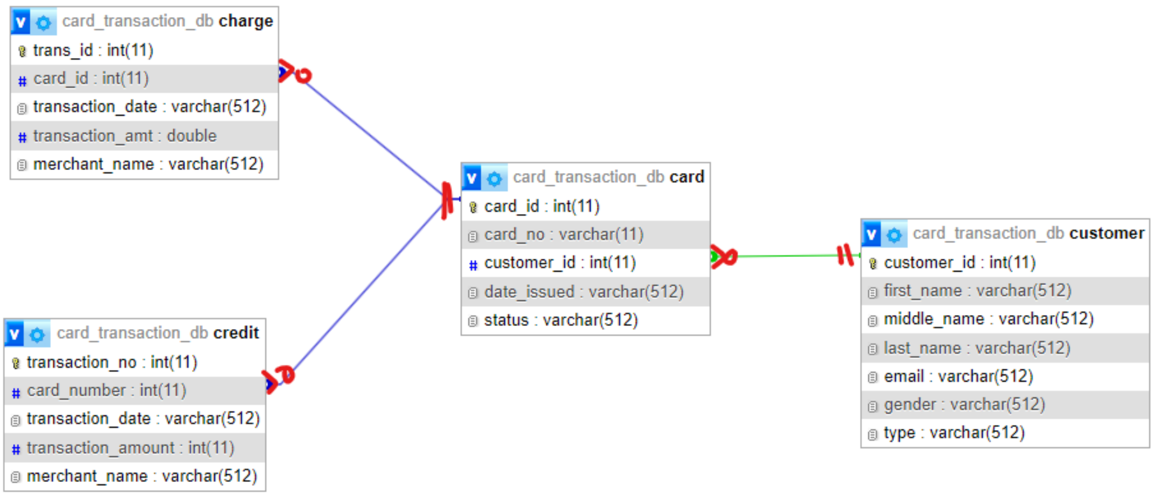
Opening other software or applications aside from mention must require permission from the instructor. Failure to do so, would result in the student forfeiting the exam and given a grade of 0 plus possible sanctions based on the student handbook for cheating.

While waiting for the exam to start please, check that you have access to the internet and the required applications.

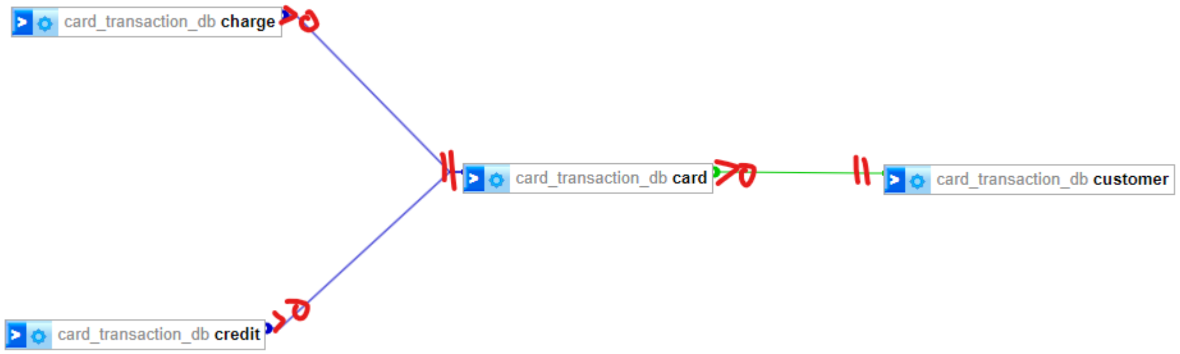
Important: **YOU WILL NOT BE ALLOWED TO RETURN TO THE PREVIOUS QUESTION whether you do it accidentally moved it or not. So please be careful when navigating your canvas accounts.**

Question Set 1: Download the following excel file ( ) as a reference for your card transactions database. Note that SET 1 does affect the set of questions in SET 2.

1. Write the ERD diagram that would represent the database provided. It should contain the following: Entities, relationship and appropriate cardinalities.



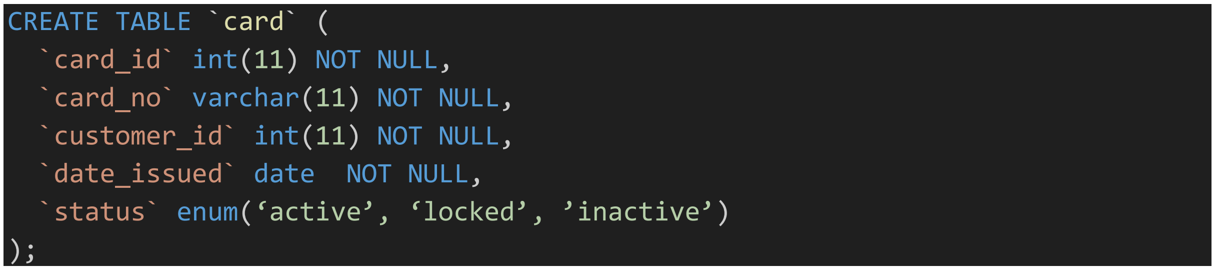
Or


card_transaction_db charge
card_transaction_db card
card_transaction_db customer
card_transaction_db credit

Checking guide:

- Entity with cardinality is okay. Entity with attributes is okay.
- Deduct 2 points per wrong relation / cardinality/ missing entity.
- Minimum score with answer 3.

2. Write the SQL command to accurately define the table for CARD. Do not include the data in the SQL definition, only the structure. Assume that other tables are also defined prior.



```
CREATE TABLE `card` (
  `card_id` int(11) NOT NULL,
  `card_no` varchar(11) NOT NULL,
  `customer_id` int(11) NOT NULL,
  `date_issued` date  NOT NULL,
  `status` enum(‘active’, ‘locked’, ’inactive’)
);
```

Checking guide:

- Full point if correct.
- Deduct 1 point per wrong line based on the guide.
- Deduct 1 pt for missing semicolon.
- Minimum score with answer -> 1.

3. Modify table customer to add a “status” column that would allow the following values: “active, inactive, and locked”. The default value for the column should be “active”.

ALTER TABLE customer

ADD status ENUM (“active”, “inactive”, “ locked”)

DEFAULT “active”;

Checking guide:

- Full point if correct.
- Deduct 1 point wrong line based on the guide
- Deduct 1 point for missing comma or semi colon
- Minimum score as long as answer is provided 1.

4. Delete the table CREDIT.

DROP TABLE credit;

Checking guide:

- Right or wrong.
- Deduct 1 point for missing comma or semi colon

Question Set 2: Download this file and upload CARD TRANSACTION DATABASE to your local XAMPP. Note that SET 1 does affect the set of questions in SET 2.

The database allows customers to register their cards and monitor their transactions.

CUSTOMER: Stores information about customers.

CARD: Stores the card registered by your customers. Each card is identified by their card number and registered by default will be active.

CREDIT: Stores reload transactions for each card.

CHARGE: Stores the spending transactions for each card.

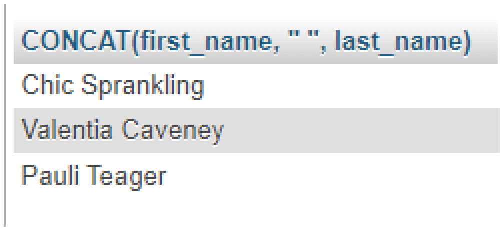
Write the SQL statements and their corresponding output. The output screenshot may only include the first 10 rows if the result is too long.

Checking guide:

- Full point if correct query and output.
- Different queries must be evaluated to see if they would yield the same result and given point accordingly.
- Deduct 1-point wrong line based on the answer provided.
- Deduct 1-point without result screenshot.
- Deduct 1 point for missing semi colon.
- Minimum score if answer is provided: 1.

1. Create list of all female student members. Only include the full name of the members.

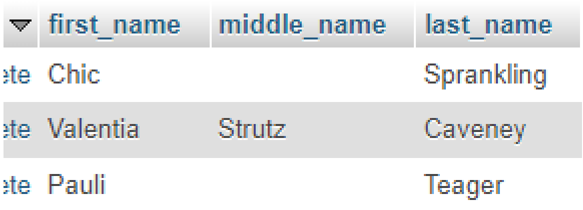
```
SELECT CONCAT(first_name, " ", last_name)
FROM customer
WHERE gender='Female'
AND type='student';
```



CONCAT(first_name, " ", last_name)
Chic Sprankling
Valentia Caveney
Pauli Teager

or

```
SELECT first_name, middle_name , last_name
FROM customer
WHERE gender='Female'
AND type='student';
```



▽	first_name	middle_name	last_name
ete	Chic		Sprankling
ete	Valentia	Strutz	Caveney
ete	Pauli		Teager

MIDDLE NAME IS OPTIONAL

Student may opt to enumerate all fields rather than * ← best practice.
Student may use alias for tables and resulting columns

2. What is the average amount spent by students, in general, per year?

DATE TYPE ERROR – NOT COUNTED – CORRECTED DB WILL BE PROVIDED FOR EXERCISE PURPOSES FOR MAJOR EXAM 2.

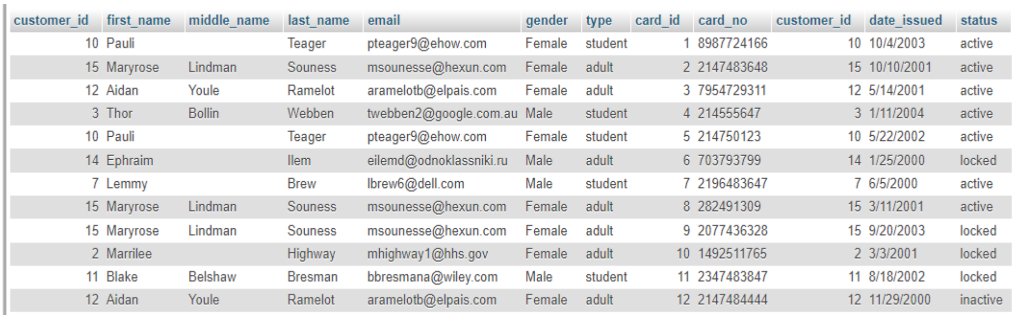
3. Show the top 10 members based on spending for the year 2021. Show the full name and the total amount spent in the said year.

DATE TYPE ERROR – NOT COUNTED – CORRECTED DB WILL BE PROVIDED FOR EXERCISE PURPOSES FOR MAJOR EXAM 2.

4. Show all the information of members who have cards. Include the entire member and card information.

```
SELECT  customer.*,  card.*
FROM customer JOIN card ON customer.customer_id=card.customer_id;

SELECT customer.*,  card.*
FROM customer, card
WHERE customer.customer_id = card.customer_id;
```

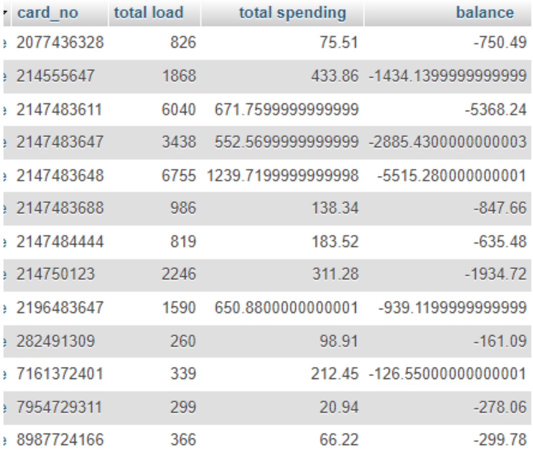


customer_id	first_name	middle_name	last_name	email	gender	type	card_id	card_no	customer_id	date_issued	status
10	Pauli		Teager	pteager9@ehow.com	Female	student	1	8987724166	10	10/4/2003	active
15	Maryrose	Lindman	Souness	msounesse@hexun.com	Female	adult	2	2147483648	15	10/10/2001	active
12	Aidan	Youle	Ramelot	aramelotb@elpais.com	Female	adult	3	7954729311	12	5/14/2001	active
3	Thor	Bollin	Webben	twebben2@google.com.au	Male	student	4	214555647	3	1/11/2004	active
10	Pauli		Teager	pteager9@ehow.com	Female	student	5	214750123	10	5/22/2002	active
14	Ephraim		Ilem	eilemd@odnoklassniki.ru	Male	adult	6	703793799	14	1/25/2000	locked
7	Lemmy		Brew	lbrew6@dell.com	Male	student	7	2196483647	7	6/5/2000	active
15	Maryrose	Lindman	Souness	msounesse@hexun.com	Female	adult	8	282491309	15	3/11/2001	active
15	Maryrose	Lindman	Souness	msounesse@hexun.com	Female	adult	9	2077436328	15	9/20/2003	locked
2	Marrilee		Highway	mhighway1@hhs.gov	Female	adult	10	1492511765	2	3/3/2001	locked
11	Blake	Belshaw	Bresman	bbresmana@wiley.com	Male	student	11	2347483847	11	8/18/2002	locked
12	Aidan	Youle	Ramelot	aramelotb@elpais.com	Female	adult	12	2147484444	12	11/29/2000	inactive

Student may opt to enumerate all fields rather than * ← best practice.
Student may use alias for tables and resulting columns

5. Show the current balance of the active cards. Show the card number, total load amount, total spending and current balance.

```
SELECT  card.card_no,
            SUM(credit.transaction_amount) AS 'total load',
            SUM(charge.transaction_amt) AS 'total spending',
            SUM(charge.transaction_amt) - SUM(credit.transaction_amount) AS 'balance'
FROM card, charge, credit
WHERE card.card_id = charge.card_id
   AND card.card_id = credit.card_number
GROUP BY card.card_no;
```

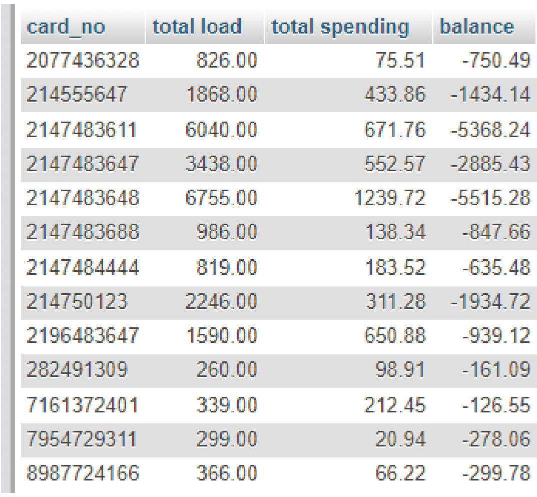


card_no	total load	total spending	balance
2077436328	826	75.51	-750.49
214555647	1868	433.86	-1434.1399999999999
2147483611	6040	671.7599999999999	-5368.24
2147483647	3438	552.5699999999999	-2885.4300000000003
2147483648	6755	1239.7199999999998	-5515.280000000001
2147483688	986	138.34	-847.66
2147484444	819	183.52	-635.48
214750123	2246	311.28	-1934.72
2196483647	1590	650.8800000000001	-939.1199999999999
282491309	260	98.91	-161.09
7161372401	339	212.45	-126.55000000000001
7954729311	299	20.94	-278.06
8987724166	366	66.22	-299.78

Student may use rounding and alias for tables and resulting columns

```
SELECT  card.card_no,
            ROUND(SUM(charge.transaction_amt),2) AS 'total load',
            ROUND(SUM(credit.transaction_amount),2) AS 'total spending',
            ROUND(SUM(charge.transaction_amt)-SUM(credit.transaction_amount),2) AS 'balance'
FROM card JOIN charge ON card.card_id = charge.card_id
             JOIN credit ON card.card_id = credit.card_number
GROUP BY card.card_no;

SELECT  card.card_no,
            ROUND(SUM(charge.transaction_amt),2) AS 'total load',
            ROUND(SUM(credit.transaction_amount),2) AS 'total spending',
            ROUND(SUM(charge.transaction_amt)-SUM(credit.transaction_amount),2) AS 'balance'
FROM card, charge, credit
WHERE card.card_id = charge.card_id
   AND card.card_id = credit.card_number
GROUP BY card.card_no;
```

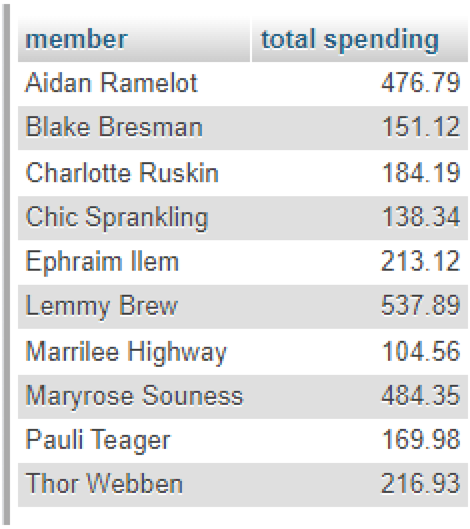


card_no	total load	total spending	balance
2077436328	826.00	75.51	-750.49
214555647	1868.00	433.86	-1434.14
2147483611	6040.00	671.76	-5368.24
2147483647	3438.00	552.57	-2885.43
2147483648	6755.00	1239.72	-5515.28
2147483688	986.00	138.34	-847.66
2147484444	819.00	183.52	-635.48
214750123	2246.00	311.28	-1934.72
2196483647	1590.00	650.88	-939.12
282491309	260.00	98.91	-161.09
7161372401	339.00	212.45	-126.55
7954729311	299.00	20.94	-278.06
8987724166	366.00	66.22	-299.78

6. Create a list of all the members who has spending transactions. Show the member name and the total amount spent. Order by name the of the members.

```
SELECT  CONCAT(customer.first_name, " ", customer.last_name) as member,
            ROUND(SUM(charge.transaction_amt),2) AS 'total spending'
FROM customer JOIN card ON customer.customer_id=card.customer_id
            JOIN charge ON card.card_id = charge.card_id
GROUP BY CONCAT(customer.first_name, " ", customer.last_name);

SELECT  CONCAT(customer.first_name, " ", customer.last_name) as member,
            ROUND(SUM(charge.transaction_amt),2) AS 'total spending'
FROM customer JOIN card ON customer.customer_id=card.customer_id
            JOIN charge ON card.card_id = charge.card_id
GROUP BY member;
```



member	total spending
Aidan Ramelot	476.79
Blake Bresman	151.12
Charlotte Ruskin	184.19
Chic Sprankling	138.34
Ephraim Ilem	213.12
Lemmy Brew	537.89
Marrilee Highway	104.56
Maryrose Souness	484.35
Pauli Teager	169.98
Thor Webben	216.93

Student may not use rounding and alias for tables and resulting columns.
Students may use manual table relation in WHERE clause instead of JOIN

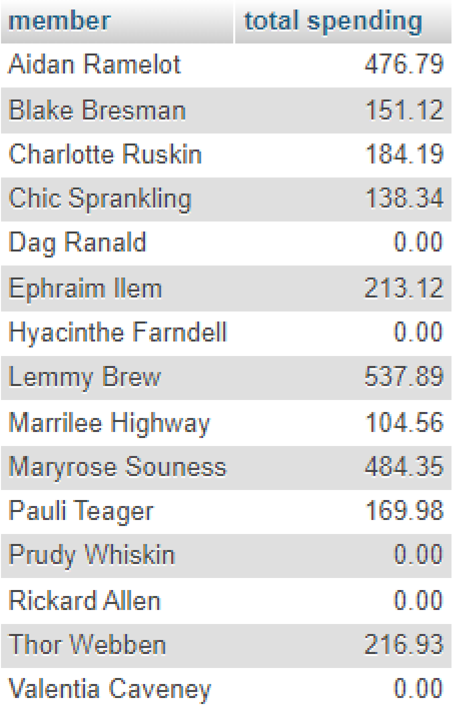
7. Show how much reload transactions were done at each merchant stores every year. Order the result by merchant name then year in ascending order.

DATE TYPE ERROR – NOT COUNTED – CORRECTED DB WILL BE PROVIDED FOR EXERCISE PURPOSES FOR MAJOR EXAM 2

8. Create a list of all the members and the total amount spent. Include those members who have not spent anything. Show the member name and the total amount spent. Order by name the of the members.

```
SELECT  CONCAT(customer.first_name, " ", customer.last_name) as member,
            ROUND(SUM(charge.transaction_amt),2) AS 'total spending'
FROM customer LEFT JOIN card ON customer.customer_id=card.customer_id
            LEFT JOIN charge ON card.card_id = charge.card_id
Group by member;
```

```
SELECT  CONCAT(customer.first_name, " ", customer.last_name) as member,
            COALESCE(ROUND(SUM(charge.transaction_amt), 2), 0) AS 'total spending'
FROM customer LEFT JOIN card ON customer.customer_id=card.customer_id
            LEFT JOIN charge ON card.card_id = charge.card_id
Group by member;
```



member	total spending
Aidan Ramelot	476.79
Blake Bresman	151.12
Charlotte Ruskin	184.19
Chic Sprankling	138.34
Dag Ranald	0.00
Ephraim Ilem	213.12
Hyacinthe Farndell	0.00
Lemmy Brew	537.89
Marrilee Highway	104.56
Maryrose Souness	484.35
Pauli Teager	169.98
Prudy Whiskin	0.00
Rickard Allen	0.00
Thor Webben	216.93
Valentia Caveney	0.00

REFER TO #6 FOR OTHER VARIATIONS AND CONSIDERATIONS

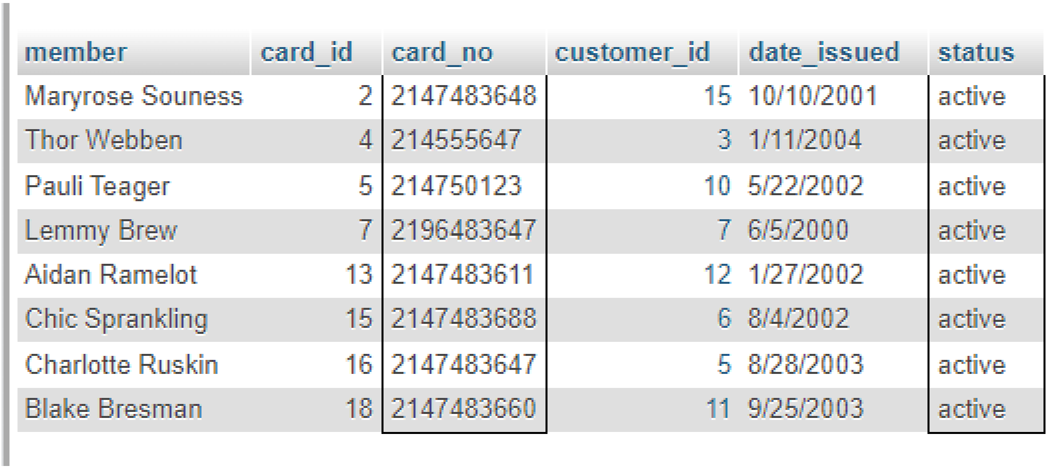
9. What is the average amount spent by adults per gender per year?

DATE TYPE ERROR – NOT COUNTED – CORRECTED DB WILL BE PROVIDED FOR EXERCISE PURPOSES FOR MAJOR EXAM 2

10. Show the list of active cards with a number starting with ‘21’. Show the member name and all card information.

Student may opt to enumerate all fields rather than * ← best practice.
Student may use alias for tables and resulting columns
Name may be concatenated or not.

```
SELECT  CONCAT(customer.first_name, " ", customer.last_name) as member,  card.*
FROM customer JOIN card ON customer.customer_id=card.customer_id
WHERE card.card_no LIKE'21%'
        AND card.status='active';
```



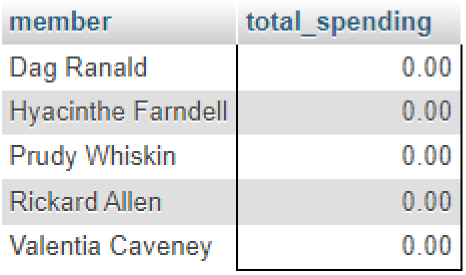
member	card_id	card_no	customer_id	date_issued	status
Maryrose Souness	2	2147483648	15	10/10/2001	active
Thor Webben	4	214555647	3	1/11/2004	active
Pauli Teager	5	214750123	10	5/22/2002	active
Lemmy Brew	7	2196483647	7	6/5/2000	active
Aidan Ramelot	13	2147483611	12	1/27/2002	active
Chic Sprankling	15	2147483688	6	8/4/2002	active
Charlotte Ruskin	16	2147483647	5	8/28/2003	active
Blake Bresman	18	2147483660	11	9/25/2003	active

11. Show all customers whose total spending is less than 100.

COALESCE DOES NOT WORK IN GROUP BY AS IT IS NOT AN AGGREGATE FUNCTION

```
SELECT member, total_spending
FROM (SELECT CONCAT(customer.first_name, " ", customer.last_name) as member,
                COALESCE(ROUND(SUM(charge.transaction_amt), 2), 0) AS total_spending
        FROM customer LEFT JOIN card ON customer.customer_id=card.customer_id
                LEFT JOIN charge ON card.card_id = charge.card_id
        GROUP BY member) AS B
 WHERE total_spending < 100;
```

```
SELECT CONCAT(customer.first_name, " ", customer.last_name) as member,
        ROUND(SUM(charge.transaction_amt), 2) AS total_spending
FROM customer JOIN card ON customer.customer_id=card.customer_id
        JOIN charge ON card.card_id = charge.card_id
GROUP BY member
HAVING SUM(charge.transaction_amt) < 100
UNION
SELECT  CONCAT(customer.first_name, " ", customer.last_name) as member, 0
FROM customer
WHERE customer_id NOT IN (
                SELECT DISTINCT customer_id
                FROM card);
```



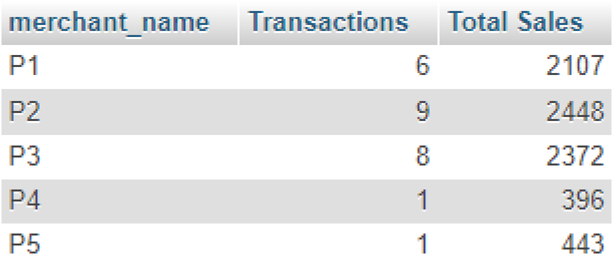
member	total_spending
Dag Ranald	0.00
Hyacinthe Farndell	0.00
Prudy Whiskin	0.00
Rickard Allen	0.00
Valentia Caveney	0.00

12. Display the number of transactions and total sales per store.

Three possible answers:

1. Sales of store based on load

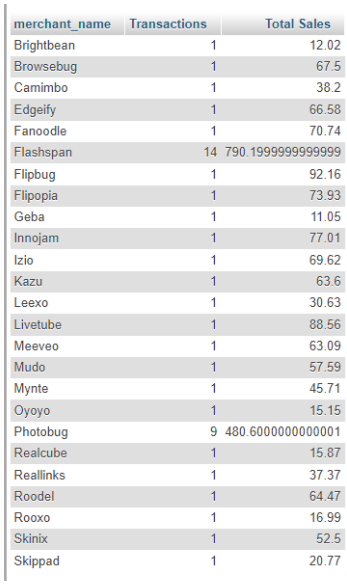
```
SELECT merchant_name, COUNT(*) as 'Transactions',
        SUM(transaction_amount) as 'Total Sales'
FROM credit
GROUP BY merchant_name;
```



merchant_name	Transactions	Total Sales
P1	6	2107
P2	9	2448
P3	8	2372
P4	1	396
P5	1	443

2. Sales of store based on spending

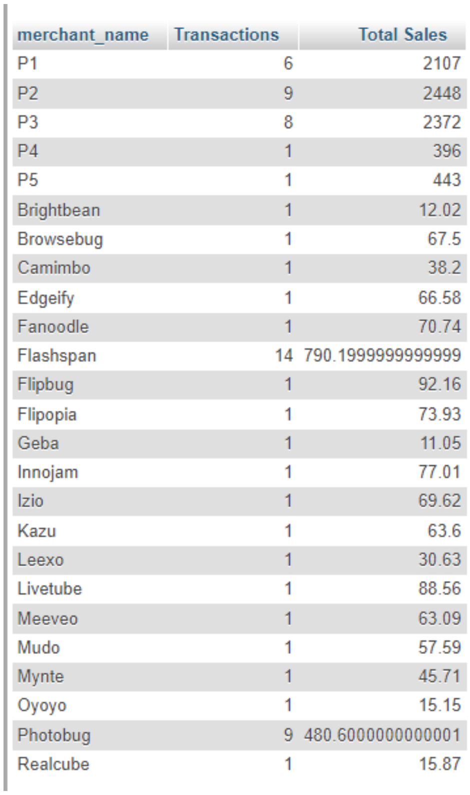
```
SELECT merchant_name, COUNT(*) as 'Transactions',
        SUM(transaction_amt) as 'Total Sales'
FROM charge
GROUP BY merchant_name;
```



merchant_name	Transactions	Total Sales
Brightbean	1	12.02
Browsebug	1	67.5
Camimbo	1	38.2
Edgeify	1	66.58
Fanoodle	1	70.74
Flashspan	14	790.1999999999999
Flipbug	1	92.16
Flipopia	1	73.93
Geba	1	11.05
Innojam	1	77.01
Izio	1	69.62
Kazu	1	63.6
Leexo	1	30.63
Livetube	1	88.56
Meeveo	1	63.09
Mudo	1	57.59
Mynte	1	45.71
Oyoyo	1	15.15
Photobug	9	480.6000000000001
Realcube	1	15.87
Reallinks	1	37.37
Roodel	1	64.47
Rooxo	1	16.99
Skinix	1	52.5
Skippad	1	20.77

3. Sales of all stores based on load and spending

```
SELECT merchant_name, COUNT(*) as 'Transactions',
        SUM(transaction_amount) as 'Total Sales'
FROM credit
GROUP BY merchant_name
UNION
SELECT merchant_name, COUNT(*) as 'Transactions',
        SUM(transaction_amt) as 'Total Sales'
FROM charge
GROUP BY merchant_name;
```



merchant_name	Transactions	Total Sales
P1	6	2107
P2	9	2448
P3	8	2372
P4	1	396
P5	1	443
Brightbean	1	12.02
Browsebug	1	67.5
Camimbo	1	38.2
Edgeify	1	66.58
Fanoodle	1	70.74
Flashspan	14	790.1999999999999
Flipbug	1	92.16
Flipopia	1	73.93
Geba	1	11.05
Innojam	1	77.01
Izio	1	69.62
Kazu	1	63.6
Leexo	1	30.63
Livetube	1	88.56
Meeveo	1	63.09
Mudo	1	57.59
Mynte	1	45.71
Oyoyo	1	15.15
Photobug	9	480.6000000000001
Realcube	1	15.87

13. What is the average amount spent by each type of customer per year?

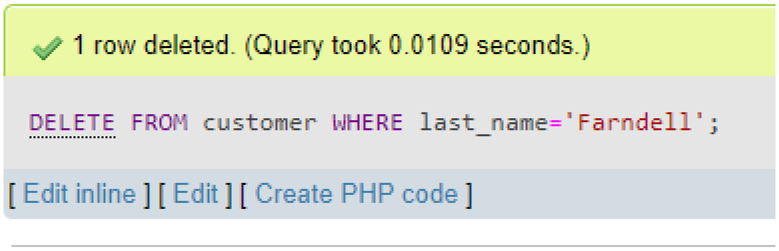
DATE TYPE ERROR – NOT COUNTED – CORRECTED DB WILL BE PROVIDED FOR EXERCISE PURPOSES FOR MAJOR EXAM 2

14. A student named Johan Merkell used his card, with a card no. of 1234987 and issued July 5, 2022, to reload. The transaction amount was 500 and was done at merchant P6. His email address was johan@mail.com.

DATE TYPE ERROR – NOT COUNTED – CORRECTED DB WILL BE PROVIDED FOR EXERCISE PURPOSES FOR MAJOR EXAM 2

15. Delete the customer record of Farndell.

```
DELETE FROM customer
WHERE last_name='Farndell';
```



✓ 1 row deleted. (Query took 0.0109 seconds.)

```
DELETE FROM customer WHERE last_name='Farndell';
```