PROJECT

PHASE – 2

CSE 5330 -005 **DATABASE SYSTEMS-I**

Prof: Bhanu Jain
GTA: Rana, Sanjivni

Team-13
Sai Krishna Prateek Nama 1001880903
Shubash Muniyappa 1001915563

PART -1

Statements to create **CUSTOMER** Table:

```
CREATE TABLE CUSTOMER (
  cId INT NOT NULL CHECK (cId > 0),
  Cname VARCHAR(255) NOT NULL,
  Street VARCHAR(255) NOT NULL,
  City VARCHAR(255) NOT NULL,
  StateAb VARCHAR(2) NOT NULL,
  Zipcode VARCHAR(10) NOT NULL,
  CONSTRAINT PK_CUSTOMER PRIMARY KEY (cId)
);
```

SELECT * FROM CUSTOMER;

Showing rows 0 - 20 (21 total, Query took 0.0005 seconds.)

```
SELECT * FROM CUSTOMER;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

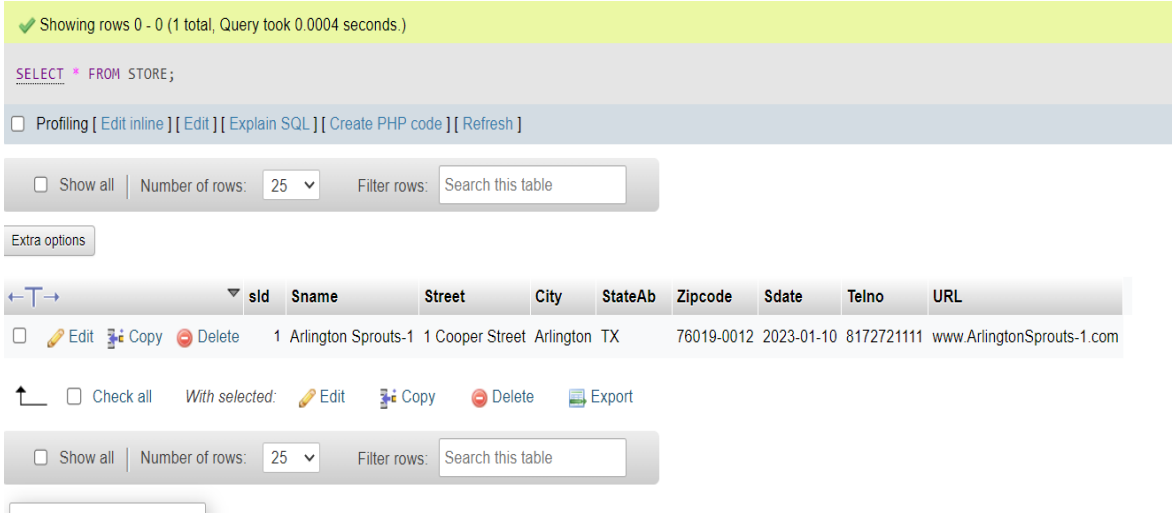
Extra options

←T→	cId	Cname	Street	City	StateAb	Zipcode
Edit Copy Delete	1	Abed Abdi	309 Hcounty Road	Abbott	TX	76621-0057
Edit Copy Delete	2	Ismail Gulg	405 E Mesquite Street	Abbott	TX	76621-0057
Edit Copy Delete	3	Shakir Ali	5000 Spectrum Street	Addison	TX	75001-6880
Edit Copy Delete	4	Abdur Rahman	1000 County Road	Bradley	Ok	73011-0121
Edit Copy Delete	5	Kalipada Ghoshal	Boundary Road	Bradley	OK	73011-9600
Edit Copy Delete	6	Manishi Dey	450 Main Street	Mound City	KS	66056-0001
Edit Copy Delete	7	Nandalal Bose	S Metcalf Road	Louisburg	KS	66053-0541
Edit Copy Delete	8	Raja Ravi Varma	2000 Forest Grove blvd	Allen	TX	75002-8811
Edit Copy Delete	9	Sunil Das	4000 Woodcreek Road	Carrollton	TX	75006-1911
Edit Copy Delete	10	Jasper Johns	2500 Sunset Ridge Drive	Rockwall	TX	75032-0006
Edit Copy Delete	11	Winslow Homer	11300 Juniper Lane	Irving	TX	75039-0101
Edit Copy Delete	12	Albert Bierstadt	400 Spruce Street	Leavenworth	KS	66048-0001
Edit Copy Delete	13	Edward Hopper	1500 255th Street	Hillsdale	KS	66036-0061

Statements to create **STORE** Table:

```
CREATE TABLE STORE (
  sId INT NOT NULL CHECK (sId > 0),
  Sname VARCHAR(255) NOT NULL,
  Street VARCHAR(255) NOT NULL,
  City VARCHAR(255) NOT NULL,
  StateAb VARCHAR(2) NOT NULL,
  Zipcode VARCHAR(10) NOT NULL,
  Sdate DATE Not NULL,
  Telno VARCHAR(15) NOT NULL,
  URL VARCHAR(255) NOT NULL,
  CONSTRAINT PK_STORE PRIMARY KEY (sId)
);
```

```
SELECT * FROM STORE;
```



Statements to create **VENDOR** Table:

```
CREATE TABLE VENDOR (
  vId INT NOT NULL CHECK (vId > 0),
  Vname VARCHAR(255) NOT NULL,
  Street VARCHAR(255)NOT NULL,
  City VARCHAR(255) NOT NULL,
  StateAb VARCHAR(2) NOT NULL,
  Zipcode VARCHAR(10) NOT NULL,
  CONSTRAINT PK_VENDOR PRIMARY KEY (vId)
);
```

```
SELECT * FROM VENDOR;
```

Showing rows 0 - 5 (6 total, Query took 0.0006 seconds.)

```
SELECT * FROM VENDOR;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→	vId	Vname	Street	City	StateAb	Zipcode
Edit Copy Delete	1	Organic Nature	1500 E Exchange Road	Allen	TX	75002-4504
Edit Copy Delete	2	Green Valley	1405 Julian Street	Addison	TX	75001-4633
Edit Copy Delete	3	Green Mountains	800 Preston Road	Gunter	TX	75058-0001
Edit Copy Delete	4	Whole Foods	1010 Spicewood Drive	Garland	TX	75044-2569
Edit Copy Delete	5	LA Queen	500 Angi Road	Minden	LA	71055-1004
Edit Copy Delete	6	Castor Sprouts	100 New Ramah Road	Castor	LA	71016-2402

Check all *With selected:* Edit Copy Delete Export

Statements to create **STORE_CUSTOMER** Table:

```
CREATE TABLE STORE_CUSTOMER (
   sId INT NOT NULL CHECK (sId > 0),
   cId INT NOT NULL CHECK (cId > 0),
   CONSTRAINT PK_STORE_CUSTOMER PRIMARY KEY (sId, cId),
   CONSTRAINT FK_STORE_STOREID FOREIGN KEY (sId) REFERENCES STORE(sId),
   CONSTRAINT FK_STORE_CUSTOMERID FOREIGN KEY (cId) REFERENCES CUSTOMER(cId)
   );
```

```
SELECT * FROM STORE_CUSTOMER;
```

Showing rows 0 - 20 (21 total, Query took 0.0006 seconds.)

```
SELECT * FROM STORE_CUSTOMER;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table S

Extra options

←T→				sId	cId
Edit	Copy	Delete		1	1
Edit	Copy	Delete		1	2
Edit	Copy	Delete		1	3
Edit	Copy	Delete		1	4
Edit	Copy	Delete		1	5
Edit	Copy	Delete		1	6
Edit	Copy	Delete		1	7
Edit	Copy	Delete		1	8
Edit	Copy	Delete		1	9
Edit	Copy	Delete		1	10
Edit	Copy	Delete		1	11
Edit	Copy	Delete		1	12
Edit	Copy	Delete		1	13

Statements to create **ITEM** Table:

```
CREATE TABLE ITEM (
  iId INT NOT NULL CHECK (iId > 0),
  Iname VARCHAR(255) NOT NULL,
  Sprice DECIMAL(10,2) NOT NULL CHECK (Sprice > 0),
  CONSTRAINT PK_ITEM PRIMARY KEY(iId)
);
```

```
SELECT * FROM ITEM;
```

Showing rows 0 - 13 (14 total, Query took 0.0006 seconds.)

```
SELECT * FROM ITEM;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

←T→	iId	Iname	Sprice
Edit Copy Delete	1	Broccoli Sprouts	3.00
Edit Copy Delete	2	Kale Sprouts	3.00
Edit Copy Delete	3	Alfalfa Sprouts	3.00
Edit Copy Delete	4	Mung Sprouts	4.00
Edit Copy Delete	5	Chickpeas Sprouts	4.00
Edit Copy Delete	6	Onion Sprouts	4.00
Edit Copy Delete	7	Soyabean Sprouts	5.00
Edit Copy Delete	8	Clover Sprouts	3.00
Edit Copy Delete	9	Kidney Beans Sprouts	4.00
Edit Copy Delete	10	Adzuki Bean Sprouts	4.00
Edit Copy Delete	11	Beet Sprouts	5.00
Edit Copy Delete	12	Green Pea Sprouts	5.00
Edit Copy Delete	13	Radish Sprouts	6.00

Statements to create **VENDOR_ITEM** Table:

```
CREATE TABLE VENDOR_ITEM (
    Vid INT NOT NULL CHECK (vId > 0),
    iId INT NOT NULL CHECK (iId > 0),
    Vprice DECIMAL (10, 2) NOT NULL,
    CONSTRAINT PK_VENDOR_ITEM PRIMARY KEY (Vid, iId),
    CONSTRAINT FK_VENDOR_ITEM FOREIGN KEY (Vid) REFERENCES VENDOR (Vid),
    CONSTRAINT FK_VENDOR_ITEMID FOREIGN KEY (iId) REFERENCES ITEM (iId)
    );
```

```
SELECT * FROM
VENDOR_ITEM;
```

Showing rows 0 - 23 (24 total, Query took 0.0005 seconds.)

```
SELECT * FROM VENDOR_ITEM;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	Vid	iId	Vprice
Edit Copy Delete	1	1	2.00
Edit Copy Delete	1	2	2.00
Edit Copy Delete	1	3	1.00
Edit Copy Delete	1	14	1.00
Edit Copy Delete	2	4	3.00
Edit Copy Delete	2	5	2.00
Edit Copy Delete	2	6	2.00
Edit Copy Delete	2	13	4.00
Edit Copy Delete	3	7	3.00
Edit Copy Delete	3	8	2.00
Edit Copy Delete	3	9	2.00
Edit Copy Delete	3	10	2.00

Statements to create **VENDOR_STORE** Table:

```
CREATE TABLE VENDOR_STORE (
   vId INT NOT NULL CHECK (vId > 0),
   sId INT NOT NULL CHECK (sId > 0),
   CONSTRAINT PK_VENDOR_STORE PRIMARY KEY (Vid, sId),
   CONSTRAINT PK_VENDOR_VENDORID FOREIGN KEY (Vid) REFERENCES VENDOR (vId),
   CONSTRAINT PK_VENDOR_VENDOR_STOREID FOREIGN KEY (Sid) REFERENCES STORE (sId)
   );
```

```
SELECT * FROM VENDOR_STORE;
```

Showing rows 0 - 5 (6 total, Query took 0.0006 seconds.)

```
SELECT * FROM VENDOR_STORE;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→			vId	sId
Edit	Copy	Delete	1	1
Edit	Copy	Delete	2	1
Edit	Copy	Delete	3	1
Edit	Copy	Delete	4	1
Edit	Copy	Delete	5	1
Edit	Copy	Delete	6	1

Check all *With selected:* Edit Copy Delete Export

Statements to create **OLDPRICE** Table:

```
CREATE TABLE OLDPRICE(
    iId INT NOT NULL CHECK (iId > 0),
    Sprice DECIMAL (10, 2) NOT NULL CHECK (Sprice >= 0),
    Sdate DATE NOT NULL,
    Edate DATE CHECK (Sdate < Edate),
    CONSTRAINT PK_OLDPRICE PRIMARY KEY (iId, Sprice),
    CONSTRAINT FK_OLDPRICE FOREIGN KEY (iId) REFERENCES ITEM (iId)
    );
```

```
SELECT * FROM OLDPRICE;
```

Showing rows 0 - 17 (18 total, Query took 0.0005 seconds.)

```
SELECT * FROM OLDPRICE;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→	iId	Sprice	Sdate	Edate
Edit Copy Delete	1	3.00	2023-01-20	NULL
Edit Copy Delete	2	3.00	2023-01-20	NULL
Edit Copy Delete	3	1.00	2023-01-20	2023-01-25
Edit Copy Delete	3	2.00	2023-01-26	2023-01-28
Edit Copy Delete	3	3.00	2023-01-29	NULL
Edit Copy Delete	4	4.00	2023-01-20	NULL
Edit Copy Delete	5	4.00	2023-01-20	NULL
Edit Copy Delete	6	4.00	2023-01-20	NULL
Edit Copy Delete	7	5.00	2023-01-20	NULL
Edit Copy Delete	8	3.00	2023-01-20	NULL
Edit Copy Delete	9	4.00	2023-01-20	NULL
Edit Copy Delete	10	4.00	2023-01-20	NULL
Edit Copy Delete	11	5.00	2023-01-20	NULL

Statements to create **CONTRACT** Table:

```
CREATE TABLE CONTRACT (
    vId INT NOT NULL CHECK (vId > 0),
    ctId INT NOT NULL CHECK (ctId BETWEEN 1 AND 10),
    Sdate DATE,
    CTime TIME,
    Ctname VARCHAR(255) NOT NULL,
    CONSTRAINT PK_CONTRACT PRIMARY KEY (vId, ctID),
    CONSTRAINT FK_CONTRACT_VENDOR FOREIGN KEY (vId) REFERENCES VENDOR (vId)
    );
```

```
SELECT * FROM CONTRACT;
```

```
SELECT * FROM `CONTRACT`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort b

+ Options

			vId	ctId	Sdate	Ctime	Cname
Edit	Copy	Delete	2	1	2024-06-24	13:00:00	Sprout Buddies
Edit	Copy	Delete	2	2	2024-07-01	13:00:00	Sprout Buddies
Edit	Copy	Delete	2	3	2024-07-07	13:00:00	Sprout Buddies
Edit	Copy	Delete	2	4	2024-07-14	14:00:00	Sprout Sandwiches
Edit	Copy	Delete	3	1	2024-06-25	15:00:00	Sprout Greens
Edit	Copy	Delete	4	1	2024-06-25	16:00:00	Sprout Certifications
Edit	Copy	Delete	4	2	2024-06-26	16:00:00	Sprout Certifications
Edit	Copy	Delete	5	1	2024-07-26	13:00:00	Sprout Certifications
Edit	Copy	Delete	5	2	2024-07-26	14:00:00	Sprout Buddies
Edit	Copy	Delete	5	3	2024-07-26	15:00:00	Sprout Sandwiches
Edit	Copy	Delete	5	4	2024-07-26	16:00:00	Sprout Sandwiches
Edit	Copy	Delete	5	5	2024-07-26	17:00:00	Microgreens

Statements to create **STORE_ITEM** Table:

```
CREATE TABLE STORE_ITEM (
    sId INT NOT NULL CHECK (sId > 0),
    iId INT NOT NULL CHECK (iId > 0),
    Scount INT CHECK (Scount >= 0),
    CONSTRAINT PK_STORE_ITEM PRIMARY KEY(sId, iID),
    CONSTRAINT FK_STOREID FOREIGN KEY (sId) REFERENCES STORE (sId),
    CONSTRAINT FK_ITEMID FOREIGN KEY (iId) REFERENCES ITEM (iId)
);

SELECT * FROM STORE_ITEM;
```

Showing rows 0 - 13 (14 total, Query took 0.0006 seconds.)

```
SELECT * FROM STORE_ITEM;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→	sId	iId	Scount
Edit Copy Delete	1	1	10
Edit Copy Delete	1	2	10
Edit Copy Delete	1	3	10
Edit Copy Delete	1	4	10
Edit Copy Delete	1	5	10
Edit Copy Delete	1	6	10
Edit Copy Delete	1	7	10
Edit Copy Delete	1	8	10
Edit Copy Delete	1	9	10
Edit Copy Delete	1	10	12
Edit Copy Delete	1	11	13
Edit Copy Delete	1	12	10
Edit Copy Delete	1	13	16

Statements to create **EMPLOYEE** Table:

```
CREATE TABLE EMPLOYEE (
   sId INT NOT NULL CHECK (sId > 0),
   SSN CHAR(9) NOT NULL,
   Ename VARCHAR(255) NOT NULL,
   Street VARCHAR (255) NOT NULL,
   City VARCHAR (255) NOT NULL,
   StateAb CHAR (2) NOT NULL,
   ZipCode INT NOT NULL,
   EType ENUM('F', 'T', 'I') NOT NULL,
   Bdate DATE NOT NULL,
   Sdate DATE NOT NULL,
   Edate DATE CHECK (EDate IS NULL OR (EDate >= SDate)),
   Levels VARCHAR(255),
   Asalary DECIMAL(10, 2),
   Agency VARCHAR (255),
   Hsalary DECIMAL (10, 2),
   Institute VARCHAR (255),
   Itype VARCHAR(10),
   CONSTRAINT PK_EMPLOYEE_SSN PRIMARY KEY(SSN),
   CONSTRAINT FK_STORE_ID FOREIGN KEY (sId) REFERENCES STORE(sId)
   );
```

Once created table, we have added a trigger to check for only 25 employees per store.

```
  DELIMITER $$
CREATE TRIGGER NUM_EMPLOYEES_STORE
        BEFORE INSERT ON EMPLOYEE
        FOR EACH ROW BEGIN
        IF (SELECT COUNT(*) FROM EMPLOYEE WHERE sId = New.sId) >= 25 THEN
        SIGNAL SQLSTATE '02000' SET MESSAGE_TEXT = 'More than 25 employees cannot work in a
store';
        END IF;
END$$
DELIMITER ;
```

SELECT * FROM EMPLOYEE;

Showing rows 0 - 5 (6 total, Query took 0.0006 seconds.)

```
SELECT * FROM EMPLOYEE;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→	sId	SSN	Ename	Street	City	StateAb	ZipCode	EType	Bdate	Sdate	Edate	Levels	Asalary	Agency	Hsalary	Institute
Edit Copy Delete	1	123456666	Olampiosi Pi	100 Varsity Circle	Arlington	TX	76019	T	2005-02-01	2023-01-20	NULL	NULL	0.00	Top Performers	10.00	NULL
Edit Copy Delete	1	123456777	Radha Gupta	200 Nedderman Dr	Arlington	TX	76019	T	2005-01-20	2023-01-20	NULL	NULL	0.00	AR Manpower	10.00	NULL
Edit Copy Delete	1	123456780	Cao Peng	12 W. Mitchell St.	Arlington	TX	76019	I	2008-01-20	2023-01-20	NULL	NULL	NULL	NULL	NULL	UTA
Edit Copy Delete	1	123456781	Giun Seong	12 S. Pecan St.	Arlington	TX	76019	I	2007-01-01	2023-01-20	NULL	NULL	NULL	NULL	NULL	Arlington High School
Edit Copy Delete	1	123456788	Christian Cart	20 Main Street	Arlington	TX	76019	F	2001-01-02	2023-01-20	NULL	Associate 1	25000.00	NULL	NULL	NULL
Edit Copy Delete	1	123456789	Mary Durer	11 Cooper Street	Arlington	TX	76019	F	2000-01-20	2023-01-20	NULL	Associate 2	30000.00	NULL	NULL	NULL

Statements to create **ORDERS** Table:

```
CREATE TABLE ORDERS (
    oId INT NOT NULL CHECK (oId > 0),
    sId INT NOT NULL CHECK (sId > 0),
    cId INT NOT NULL CHECK (cId > 0),
    Odate DATE NOT NULL,
    Ddate DATE NOT NULL CHECK (Ddate >= Odate),
    Amount INT NOT NULL CHECK (Amount > 0),
    PRIMARY KEY (oId),
    CONSTRAINT FK_sId FOREIGN KEY (sId) REFERENCES STORE (sId),
    CONSTRAINT FK_cId FOREIGN KEY (cId) REFERENCES CUSTOMER (cId)
    );
```

```
SELECT * FROM ORDERS;
```

Showing rows 0 - 24 (27 total, Query took 0.0005 seconds.)

```
SELECT * FROM ORDERS;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

←T→	oId	sId	cId	Odate	Ddate	Amount
Edit Copy Delete	1	1	1	2023-01-20	2023-01-21	3
Edit Copy Delete	2	1	2	2023-01-20	2023-01-21	3
Edit Copy Delete	3	1	3	2023-01-20	2023-01-21	4
Edit Copy Delete	4	1	4	2023-01-20	2023-01-21	12
Edit Copy Delete	5	1	5	2023-01-20	2023-01-21	4
Edit Copy Delete	6	1	6	2023-01-20	2023-01-21	27
Edit Copy Delete	7	1	7	2023-01-20	2023-01-21	4
Edit Copy Delete	8	1	8	2023-01-20	2023-01-21	5
Edit Copy Delete	9	1	9	2023-01-20	2023-01-21	5
Edit Copy Delete	10	1	10	2023-01-20	2023-01-21	5
Edit Copy Delete	11	1	11	2023-01-20	2023-01-21	3
Edit Copy Delete	12	1	12	2023-01-25	2023-01-26	4
Edit Copy Delete	13	1	13	2023-01-25	2023-01-26	4

Statements to create **ORDER_ITEM** Table:

```
CREATE TABLE ORDER_ITEM(
  oId INT NOT NULL CHECK (oId > 0),
   sId INT NOT NULL CHECK (sId > 0),
   iId INT NOT NULL CHECK (iId > 0),
   Icount INT,
   CONSTRAINT PK_ORDER PRIMARY KEY (oId, sId, iId),
   CONSTRAINT FK_ORDER FOREIGN Key (oId) REFERENCES ORDERS (oId),
   CONSTRAINT FK_STORE FOREIGN Key (sId) REFERENCES STORE (sId),
   CONSTRAINT FK_ITEM FOREIGN Key (iId) REFERENCES ITEM (iId)
);
```

```
SELECT * FROM ORDER_ITEM;
```

Showing rows 0 - 24 (33 total, Query took 0.0008 seconds.)

```
SELECT * FROM ORDER_ITEM;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

				oId	sId	iId	Icount
Edit	Copy	Delete		1	1	1	1
Edit	Copy	Delete		2	1	2	1
Edit	Copy	Delete		3	1	4	1
Edit	Copy	Delete		4	1	5	1
Edit	Copy	Delete		4	1	6	2
Edit	Copy	Delete		5	1	5	1
Edit	Copy	Delete		6	1	5	1
Edit	Copy	Delete		6	1	6	2
Edit	Copy	Delete		6	1	7	3
Edit	Copy	Delete		7	1	6	1
Edit	Copy	Delete		8	1	7	1
Edit	Copy	Delete		9	1	7	1

PART -2

Q1. Retrieve the names and the address of all the customers **whose last names** starts with a 'J' or 'M'.

SELECT Cname, Street, City, StateAb,Zipcode FROM`CUSTOMER` where Cname LIKE '% J%' OR Cname LIKE '% M%';

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

```
SELECT Cname, Street, City, StateAb,Zipcode FROM`CUSTOMER` where Cname LIKE '% J%' OR Cname LIKE '% M%';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	Cname	Street	City	StateAb	Zipcode
Edit Copy Delete	Jasper Johns	2500 Sunset Ridge Drive	Rockwall	TX	75032-0006
Edit Copy Delete	Ekene Maduka	15TH Street	Plano	TX	75086-0015
Edit Copy Delete	Tatsuo Miyajima	Bella Ranch Drive	Choctaw	OK	73020-0017

Check all *With selected:* Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Q2. State the names of vendors, the courses that they offer, the date, and the time at which they offer the **courses after 2:00 PM.** Display the time in **12-hour AM/PM format.**

SELECT `VENDOR`.`Vname` as 'Vendor Name', `CONTRACT`.`Ctname` as 'Course Name', `CONTRACT`.`Sdate` AS 'Course Date', TIME_FORMAT(`CONTRACT`.`CTime`, '%r') As 'Course Time' FROM `VENDOR` LEFT JOIN `CONTRACT` ON `CONTRACT`.`vId` = `VENDOR`.`vId` WHERE `CONTRACT`.`CTime` > '14:00';

Showing rows 0 - 5 (6 total, Query took 0.0088 seconds.)

```
SELECT `VENDOR`.`Vname` as 'Vendor Name', `CONTRACT`.`Cname` as 'Course Name', `CONTRACT`.`Sdate` AS 'Course Date',
TIME_FORMAT(`CONTRACT`.`Ctime`, '%r') As 'Course Time' FROM `VENDOR` LEFT JOIN `CONTRACT` ON `CONTRACT`.`vId` = `VENDOR`.`vId` WHERE
`CONTRACT`.`Ctime` > '14:00';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

Vendor Name	Course Name	Course Date	Course Time
Green Mountains	Sprout Greens	2024-06-25	03:00:00 PM
Whole Foods	Sprout Certifications	2024-06-25	04:00:00 PM
Whole Foods	Sprout Certifications	2024-06-26	04:00:00 PM
LA Queen	Sprout Sandwiches	2024-07-26	03:00:00 PM
LA Queen	Sprout Sandwiches	2024-07-26	04:00:00 PM
LA Queen	Microgreens	2024-07-26	05:00:00 PM

Q3. Retrieve the names of the vendors, the items that they supply, and the price at which they supply these at. Additionally, list the price at which Arlington Sprouts sell these items.
The dollar amount should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

```
SELECT `VENDOR`.`Vname`, `ITEM`.`Iname`, concat('$',`VENDOR_ITEM`.`Vprice`) as 'Vendor Price', concat('$',`ITEM`.`Sprice`) as 'Store Price'

FROM `VENDOR`

        LEFT JOIN `VENDOR_ITEM` ON `VENDOR_ITEM`.`vId` = `VENDOR`.`vId`

        LEFT JOIN `ITEM` ON `VENDOR_ITEM`.`iId` = `ITEM`.`iId`;
```

Vname	Iname	Vendor Price	Store Price
Organic Nature	Broccoli Sprouts	$2.00	$3.00
Organic Nature	Kale Sprouts	$2.00	$3.00
Organic Nature	Alfalfa Sprouts	$1.00	$3.00
Organic Nature	Lentil Sprouts	$1.00	$4.00
Green Valley	Mung Sprouts	$3.00	$4.00
Green Valley	Chickpeas Sprouts	$2.00	$4.00
Green Valley	Onion Sprouts	$2.00	$4.00
Green Valley	Radish Sprouts	$4.00	$6.00
Green Mountains	Soyabean Sprouts	$3.00	$5.00
Green Mountains	Clover Sprouts	$2.00	$3.00
Green Mountains	Kidney Beans Sprouts	$2.00	$4.00
Green Mountains	Adzuki Bean Sprouts	$2.00	$4.00
Whole Foods	Beet Sprouts	$3.00	$5.00
Whole Foods	Green Pea Sprouts	$3.00	$5.00
Whole Foods	Radish Sprouts	$2.00	$6.00
Whole Foods	Lentil Sprouts	$2.00	$4.00
LA Queen	Soyabean Sprouts	$3.00	$5.00
LA Queen	Beet Sprouts	$2.00	$5.00
LA Queen	Radish Sprouts	$5.00	$6.00
LA Queen	Lentil Sprouts	$3.00	$4.00
Castor Sprouts	Soyabean Sprouts	$2.00	$5.00
Castor Sprouts	Clover Sprouts	$1.00	$3.00
Castor Sprouts	Kidney Beans Sprouts	$3.00	$4.00
Console uts	Adzuki Bean Sprouts	$4.00	$4.00

Q4. Retrieve the names of the 10 most expensive items sold at Arlington Sprouts. List the items in a descending order by price and the item names in an ascending order for each price point.
The dollar amount should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

SELECT `ITEM`.`Iname` as 'Item Name' , concat('$',`ITEM`.`Sprice`) as 'Store Price' FROM `ITEM` ORDER by Sprice DESC, Iname ASC LIMIT 10;

Showing rows 0 - 9 (10 total, Query took 0.0006 seconds.)

```
SELECT `ITEM`.`Iname` as 'Item Name' , concat('$',`ITEM`.`Sprice`) as 'Store Price' FROM `ITEM` ORDER by Sprice DESC, Iname ASC LIMIT 10;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Extra options

←T→	Item Name	Store Price
Edit Copy Delete	Radish Sprouts	$6.00
Edit Copy Delete	Beet Sprouts	$5.00
Edit Copy Delete	Green Pea Sprouts	$5.00
Edit Copy Delete	Soyabean Sprouts	$5.00
Edit Copy Delete	Adzuki Bean Sprouts	$4.00
Edit Copy Delete	Chickpeas Sprouts	$4.00
Edit Copy Delete	Kidney Beans Sprouts	$4.00
Edit Copy Delete	Lentil Sprouts	$4.00
Edit Copy Delete	Mung Sprouts	$4.00
Edit Copy Delete	Onion Sprouts	$4.00

Check all *With selected:* Edit Copy Delete Export

Q5. Retrieve the vendor names, item names, vendor prices, and the store prices when the store charges **more than double the** vendor price.
The dollar amount should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

SELECT `VENDOR`.`Vname`,`ITEM`.`Iname`, concat('$',`VENDOR_ITEM`.`Vprice`) as 'Vendor Price', concat('$',`ITEM`.`Sprice`) as 'Store Price' FROM `VENDOR` LEFT JOIN `VENDOR_ITEM` ON `VENDOR_ITEM`.`vId` = `VENDOR`.`vId` LEFT JOIN `ITEM` ON `VENDOR_ITEM`.`iId` = `ITEM`.`iId` WHERE `ITEM`.`Sprice` > `VENDOR_ITEM`.`Vprice`*2;

Showing rows 0 - 5 (6 total, Query took 0.0008 seconds.)

```
SELECT `VENDOR`.`Vname`,`ITEM`.`Iname`, concat('$',`VENDOR_ITEM`.`Vprice`) as 'Vendor Price', concat('$',`ITEM`.`Sprice`) as 'Store Price' FROM `VENDOR` LEFT JOIN
`VENDOR_ITEM` ON `VENDOR_ITEM`.`vId` = `VENDOR`.`vId` LEFT JOIN `ITEM` ON `VENDOR_ITEM`.`iId` = `ITEM`.`iId` WHERE `ITEM`.`Sprice` > `VENDOR_ITEM`.`Vprice`*2;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

Vname	Iname	Vendor Price	Store Price
Organic Nature	Alfalfa Sprouts	$1.00	$3.00
Organic Nature	Lentil Sprouts	$1.00	$4.00
Whole Foods	Radish Sprouts	$2.00	$6.00
LA Queen	Beet Sprouts	$2.00	$5.00
Castor Sprouts	Soyabean Sprouts	$2.00	$5.00
Castor Sprouts	Clover Sprouts	$1.00	$3.00

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Q6. Retrieve the vendor names, item names, vendor prices, and the store prices when the store and the vendor prices differed by exactly 2 dollars. Order the results by Vendor Names.
The dollar amount should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

SELECT VENDOR.Vname AS 'Vendor Name', ITEM.Iname AS 'Item Name', VENDOR_ITEM.Vprice AS 'Vendor Price', ITEM.Sprice AS 'Store Price'

FROM (vendor_item

INNER JOIN vendor ON VENDOR_ITEM.Vid =VENDOR.vId)

INNER JOIN ITEM ON vendor_item.iId = ITEM.iId

WHERE (ITEM.Sprice - VENDOR_ITEM.Vprice) = 2

ORDER BY VENDOR.Vname;

Showing rows 0 - 11 (12 total, Query took 0.0008 seconds.)

```
SELECT VENDOR.Vname AS 'Vendor Name', ITEM.Iname AS 'Item Name', VENDOR_ITEM.Vprice AS 'Vendor Price', ITEM.Sprice AS 'Store Price' FROM (vendor_item INNER JOIN vendor ON
VENDOR_ITEM.Vid =VENDOR.vId) INNER JOIN ITEM ON vendor_item.iId = ITEM.iId WHERE (ITEM.Sprice - VENDOR_ITEM.Vprice) = 2 ORDER BY VENDOR.Vname;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

Vendor Name	Item Name	Vendor Price	Store Price
Castor Sprouts	Clover Sprouts	1.00	3.00
Green Mountains	Soyabean Sprouts	3.00	5.00
Green Mountains	Kidney Beans Sprouts	2.00	4.00
Green Mountains	Adzuki Bean Sprouts	2.00	4.00
Green Valley	Chickpeas Sprouts	2.00	4.00
Green Valley	Onion Sprouts	2.00	4.00
Green Valley	Radish Sprouts	4.00	6.00
LA Queen	Soyabean Sprouts	3.00	5.00
Organic Nature	Alfalfa Sprouts	1.00	3.00
Whole Foods	Beet Sprouts	3.00	5.00
Whole Foods	Green Pea Sprouts	3.00	5.00
Whole Foods	Lentil Sprouts	2.00	4.00

Show all | Number of rows: 25 Filter rows: Search this table

Console

Q7 Retrieve a list of date, customer name, the total order amount, and the number of boxes of items(sprouts) bought by them. Let the list be sorted in an ascending order by date and the customers' names.
The dollar amount should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00) .

NOTE: For the question we have two queries as we had confusion regarding the whether to display amount with respect to customer or not because Abdul Rahman has placed order twice on the same day.

```
SELECT `ORDERS`.`Odate` AS 'Order Date', `CUSTOMER`.`Cname` AS 'Customer Name',
`ORDERS`.`Amount`,  `ORDER_ITEM`.`Icount` AS 'Items Ordered'

FROM `ORDERS`

        LEFT JOIN `CUSTOMER` ON `ORDERS`.`cId` = `CUSTOMER`.`cId`

        LEFT JOIN `ORDER_ITEM` ON `ORDER_ITEM`.`oId` = `ORDERS`.`oId` ORDER BY
`ORDERS`.`Odate` ASC, `CUSTOMER`.`Cname` ASC;
```

Showing rows 0 - 24 (33 total, Query took 0.0009 seconds.)

```
SELECT `ORDERS`.`Odate` AS 'Order Date', `CUSTOMER`.`Cname` AS 'Customer Name', `ORDERS`.`Amount`, `ORDER_ITEM`.`Icount` AS 'Items Ordered' FROM `ORDERS` LEFT JOIN
`CUSTOMER` ON `ORDERS`.`cId` = `CUSTOMER`.`cId` LEFT JOIN `ORDER_ITEM` ON `ORDER_ITEM`.`oId` = `ORDERS`.`oId` ORDER BY `ORDERS`.`Odate` ASC, `CUSTOMER`.`Cname` ASC;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

Order Date	Customer Name	Amount	Items Ordered
2023-01-20	Abdur Rahman	12	1
2023-01-20	Abdur Rahman	12	2
2023-01-20	Abed Abdi	3	1
2023-01-20	Ismail Gulg	3	1
2023-01-20	Jasper Johns	5	1
2023-01-20	Kalipada Ghoshal	4	1
2023-01-20	Manishi Dey	27	1
2023-01-20	Manishi Dey	27	2
2023-01-20	Manishi Dey	27	3
2023-01-20	Nandalal Bose	4	1
2023-01-20	Raja Ravi Varma	5	1
2023-01-20	Shakir Ali	4	1
2023-01-20	Sunil Das	5	1

Order Date	Customer Name	Amount	Items Ordered
2023-01-20	Sunil Das	5	1
2023-01-20	Winslow Homer	3	1
2023-01-25	Albert Bierstadt	4	1
2023-01-25	Edward Hopper	4	1
2023-01-25	Georgia O'Keeffe	4	1
2023-01-25	Ismail Gulg	22	1
2023-01-25	Ismail Gulg	22	1
2023-01-25	Ismail Gulg	22	1
2023-01-25	Ismail Gulg	22	2
2023-01-25	Kalipada Ghoshal	5	1
2023-01-25	Sunil Das	6	1
2023-01-30	Ekene Maduka	3	1
2023-01-30	Ismail Gulg	4	1

SELECT `ORDERS`.`Odate` as 'Order date' , `CUSTOMER`.`Cname` as 'Customer Name' , CONCAT('$', FORMAT(SUM(`ORDERS`.`Amount`), 2)) AS 'Amount', SUM(`ORDER_ITEM`.`Icount`) AS 'Items Ordered' FROM `ORDERS` LEFT JOIN `CUSTOMER` ON `ORDERS`.`cId` = `CUSTOMER`.`cId` LEFT JOIN `ORDER_ITEM` ON `ORDER_ITEM`.`oId` = `ORDERS`.`oId` GROUP BY `ORDERS`.`Odate`, `CUSTOMER`.`Cname` ORDER BY `ORDERS`.`Odate` ASC, `CUSTOMER`.`Cname` ASC;

Showing rows 0 - 24 (27 total, Query took 0.0010 seconds.)

```
SELECT `ORDERS`.`Odate` as 'Order date' , `CUSTOMER`.`Cname` as 'Customer Name' , CONCAT('$', FORMAT(SUM(`ORDERS`.`Amount`), 2)) AS 'Amount', SUM(`ORDER_ITEM`.`Icount`) AS 'Items Ordered' FROM `ORDERS` LEFT JOIN `CUSTOMER` ON `ORDERS`.`cId` = `CUSTOMER`.`cId` LEFT JOIN `ORDER_ITEM` ON `ORDER_ITEM`.`oId` = `ORDERS`.`oId` GROUP BY `ORDERS`.`Odate`, `CUSTOMER`.`Cname` ORDER BY `ORDERS`.`Odate` ASC, `CUSTOMER`.`Cname` ASC;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | Show all | Number of rows: 25 Filter rows: Search this table

Extra options

Order date	Customer Name	Amount	Items Ordered
2023-01-20	Abdur Rahman	$24.00	3
2023-01-20	Abed Abdi	$3.00	1
2023-01-20	Ismail Gulg	$3.00	1
2023-01-20	Jasper Johns	$5.00	1
2023-01-20	Kalipada Ghoshal	$4.00	1
2023-01-20	Manishi Dey	$81.00	6
2023-01-20	Nandalal Bose	$4.00	1
2023-01-20	Raja Ravi Varma	$5.00	1
2023-01-20	Shakir Ali	$4.00	1
2023-01-20	Sunil Das	$5.00	1
2023-01-20	Winslow Homer	$3.00	1
2023-01-25	Albert Bierstadt	$4.00	1
2023-01-25	Edward Hopper	$4.00	1
2023-01-25	Georgia O'Keeffe	$4.00	1
Console	Ismail Gulg	$88.00	5

Order date	Customer Name	Amount	Items Ordered
2023-01-25	Ismail Gulg	$88.00	5
2023-01-25	Kalipada Ghoshal	$5.00	1
2023-01-25	Sunil Das	$6.00	1
2023-01-30	Ekene Maduka	$3.00	1
2023-01-30	Ismail Gulg	$4.00	1
2023-01-30	Modupeola Fadugba	$6.00	1
2023-01-30	Olu Amoda	$3.00	1
2023-01-30	Sunil Das	$4.00	1
2023-02-05	Koki Tanaka	$4.00	1
2023-02-05	Li Chen	$4.00	1
2023-02-05	Tatsuo Miyajima	$4.00	1

1 > >> | Show all | Number of rows: 25 Filter rows: Search this table

Q8. Retrieve, the total, average, maximum amounts of order placed in the store so far by date.
The price should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

SELECT Odate AS 'Order Date', CONCAT('$',COUNT(Amount)) AS 'Sum AMT', CONCAT('$',AVG(Amount)) AS 'Avg AMT', CONCAT('$',MAX(AMOUNT)) AS 'Max AMT' FROM ORDERS GROUP BY Odate;

Showing rows 0 - 5 (6 total, Query took 0.0004 seconds.)

```
SELECT Odate AS 'Order Date', CONCAT('$',COUNT(Amount)) AS 'Sum AMT', CONCAT('$',AVG(Amount)) AS 'Avg AMT', CONCAT('$',MAX(AMOUNT)) AS 'Max AMT' FROM ORDERS GROUP BY
Odate;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	Order Date	Sum AMT	Avg AMT	Max AMT
Edit Copy Delete	2023-01-20	$11	$6.8182	$27
Edit Copy Delete	2023-01-25	$6	$7.5000	$22
Edit Copy Delete	2023-01-30	$5	$4.0000	$6
Edit Copy Delete	2023-02-05	$3	$4.0000	$4
Edit Copy Delete	2023-02-10	$1	$4.0000	$4
Edit Copy Delete	2023-02-12	$1	$4.0000	$4

Check all *With selected:* Edit Copy Delete Export

Q9 Retrieve **a list of records** where each record is comprised of the order date, customer name, and the minimum or the max order amount. If there are multiple such orders, list the information for the minimum amount orders followed by the information for the maximum amount.
The dollar amount should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

```
SELECT o.Odate AS 'Order date', c.Cname AS 'Customer Name', CONCAT('$', FORMAT(CASE WHEN
o.Amount = min_max.min_amount THEN min_max.min_amount ELSE min_max.max_amount END, 2))
AS Amount FROM orders o
    JOIN customer c ON o.cId = c.cId
    JOIN (SELECT MIN(orders.Amount) AS min_amount,  MAX(orders.Amount) AS max_amount
        FROM orders) min_max ON o.Amount IN (min_max.min_amount, min_max.max_amount);
```

Showing rows 0 - 5 (6 total, Query took 0.0014 seconds.)

```
SELECT o.Odate AS 'Order date', c.Cname AS 'Customer Name', CONCAT('$', FORMAT(CASE WHEN o.Amount = min_max.min_amount THEN min_max.min_amount ELSE min_max.max_amount END,
2)) AS Amount FROM orders o JOIN customer c ON o.cId = c.cId JOIN ( SELECT MIN(orders.Amount) AS min_amount, MAX(orders.Amount) AS max_amount FROM orders ) min_max ON
o.Amount IN (min_max.min_amount, min_max.max_amount);
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

Order date	Customer Name	Amount
2023-01-20	Abed Abdi	$3.00
2023-01-20	Ismail Gulg	$3.00
2023-01-20	Manishi Dey	$27.00
2023-01-20	Winslow Homer	$3.00
2023-01-30	Ekene Maduka	$3.00
2023-01-30	Olu Amoda	$3.00

Show all | Number of rows: 25 Filter rows: Search this table

Q10 Retrieve **a record** which is comprised of information about the minimum and maximum amount of order placed. The record will need to list the order date, customer name, and the order amount for the minimum and maximum orders. If there are multiple customers with the minimum and/or maximum order, list only the first name in the alphabetical order of such a list.
The dollar amount should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

SELECT MIN(o.Odate) AS 'MinOrder Date', SUBSTRING_INDEX(MIN(c.Cname), ' ', 1) AS 'MinCustomer Name', CONCAT('$', FORMAT(MIN(o.Amount), 2)) AS 'MinAmount', MAX(o.Odate) AS 'MaxOrder Date', SUBSTRING_INDEX(MIN(c.Cname), ' ', 1) AS 'MaxCustomer Name', CONCAT('$', FORMAT(MAX(o.Amount), 2)) AS 'Max Amount'
FROM orders o LEFT OUTER JOIN customer c ON o.cId = c.cId
WHERE o.Amount = (SELECT MIN(Amount) FROM orders) OR o.Amount = (SELECT MAX(Amount) FROM orders);

Showing rows 0 - 0 (1 total, Query took 0.0009 seconds.)

```
SELECT MIN(o.Odate) AS 'MinOrder Date', SUBSTRING_INDEX(MIN(c.Cname), ' ', 1) AS 'MinCustomer Name', CONCAT('$', FORMAT(MIN(o.Amount), 2)) AS 'MinAmount', MAX(o.Odate) AS 'MaxOrder Date', SUBSTRING_INDEX(MIN(c.Cname), ' ', 1) AS 'MaxCustomer Name', CONCAT('$', FORMAT(MAX(o.Amount), 2)) AS 'Max Amount' FROM orders o LEFT OUTER JOIN customer c ON o.cId = c.cId WHERE o.Amount = (SELECT MIN(Amount) FROM orders) OR o.Amount = (SELECT MAX(Amount) FROM orders);
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

MinOrder Date	MinCustomer Name	MinAmount	MaxOrder Date	MaxCustomer Name	Max Amount
2023-01-20	Abed	$3.00	2023-01-30	Abed	$27.00

Show all | Number of rows: 25 Filter rows: Search this table

Q11 Retrieve **a list of records** that state the information pertaining to the dates, number of min/max orders, and the total amounts of such orders on the pertinent dates on which the minimum and maximum orders were placed. Each record in this list should be comprised of the date (when min or max amount order was placed), number of minimum or maximum orders separately, and the total of such min or max amount orders individually.
If multiple minimum and maximum orders are placed on same dates, add the min amount and max amount orders separately and list them individually.
The dollar amount should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

```
SELECT ORDERS.Odate AS Date,
CONCAT('Min:', COUNT(CASE WHEN ORDERS.Amount = min_total THEN 1 END), ', Max:',
COUNT(CASE WHEN ORDERS.Amount = max_total THEN 1 END) ) AS '#orders',
CONCAT('Min:','$', FORMAT(SUM(CASE WHEN ORDERS.Amount = min_total THEN
ORDERS.Amount END), 2), CONCAT(', Max:','$', FORMAT(SUM(CASE WHEN ORDERS.Amount =
max_total THEN ORDERS.Amount END), 2))) AS 'Amount'
FROM orders JOIN (SELECT ORDERS.Odate, MIN(ORDERS.Amount) AS min_total,
MAX(ORDERS.Amount) AS max_total FROM orders GROUP BY ORDERS.Odate) AS total ON
orders.Odate = total.Odate GROUP BY orders.Odate;
```

Showing rows 0 - 5 (6 total, Query took 0.0012 seconds.)

```
SELECT ORDERS.Odate AS Date, CONCAT('Min:', COUNT(CASE WHEN ORDERS.Amount = min_total THEN 1 END), ', Max:', COUNT(CASE WHEN ORDERS.Amount = max_total THEN 1 END) ) AS
'#orders', CONCAT('Min:','$', FORMAT(SUM(CASE WHEN ORDERS.Amount = min_total THEN ORDERS.Amount END), 2), CONCAT(', Max:','$', FORMAT(SUM(CASE WHEN ORDERS.Amount =
max_total THEN ORDERS.Amount END), 2))) AS 'Amount' FROM orders JOIN (SELECT ORDERS.Odate, MIN(ORDERS.Amount) AS min_total, MAX(ORDERS.Amount) AS max_total FROM orders
GROUP BY ORDERS.Odate) AS total ON orders.Odate = total.Odate GROUP BY orders.Odate;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

Date	#orders	Amount
2023-01-20	Min:3, Max:1	Min:$9.00, Max:$27.00
2023-01-25	Min:3, Max:1	Min:$12.00, Max:$22.00
2023-01-30	Min:2, Max:1	Min:$6.00, Max:$6.00
2023-02-05	Min:3, Max:3	Min:$12.00, Max:$12.00
2023-02-10	Min:1, Max:1	Min:$4.00, Max:$4.00
2023-02-12	Min:1, Max:1	Min:$4.00, Max:$4.00

Show all | Number of rows: 25 Filter rows: Search this table

Q12 Retrieve a list of records that lists the name of customers and the sum of all the orders that they have ever placed at any of the Arlington Sprouts store. List the records in an ascending order by the total amount spent by customers at the store. The dollar amount should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

```
SELECT `CUSTOMER`.`Cname` as 'Customer Name',CONCAT('$', FORMAT(SUM(
`ORDERS`.`Amount`), 2)) AS 'Total Amount Spent' FROM `CUSTOMER` LEFT JOIN `ORDERS` ON
`ORDERS`.`cId` = `CUSTOMER`.`cId` GROUP BY `CUSTOMER`.`Cname` ORDER BY
SUM(`ORDERS`.`Amount`) ASC;
```

Customer Name	Total Amount Spent
Abed Abdi	$3.00
Olu Amoda	$3.00
Winslow Homer	$3.00
Ekene Maduka	$3.00
Koki Tanaka	$4.00
Shakir Ali	$4.00
Tatsuo Miyajima	$4.00
Albert Bierstadt	$4.00
Li Chen	$4.00
Edward Hopper	$4.00
Georgia O'Keeffe	$4.00
Nandalal Bose	$4.00
Jasper Johns	$5.00
Raja Ravi Varma	$5.00
Modupeola Fadugba	$6.00
Zhan Wang	$8.00
Kalipada Ghoshal	$9.00
Abdur Rahman	$12.00
Sunil Das	$15.00
Manishi Dey	$27.00
Ismail Gulg	$29.00

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Console

Q13 Retrieve a list of records that lists the name of customers, the total number of items (boxes of sprouts) they ordered at any time, and the sum of all the order amounts that they have ever placed at any of the Arlington Sprouts store . List the records in an ascending order by the total number of items(boxes of sprouts) ordered at any time , followed by the total amount spent by customers at the store. The dollar amount should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

```
SELECT `CUSTOMER`.`Cname`, SUM(`ORDER_ITEM`.`Icount`) AS 'Total Boxes',CONCAT('$',
FORMAT(SUM(`ORDERS`.`Amount`), 2)) AS 'Total Amount' FROM `ORDERS` LEFT JOIN
`CUSTOMER` ON `ORDERS`.`cId` = `CUSTOMER`.`cId` LEFT JOIN `ORDER_ITEM` ON
`ORDER_ITEM`.`oId` = `ORDERS`.`oId` GROUP BY `CUSTOMER`.`Cname` ORDER BY
SUM(`ORDER_ITEM`.`Icount`) ASC, SUM(`ORDERS`.`Amount`) ASC;
```

Cname	Total Boxes	Total Amount
Abed Abdi	1	$3.00
Olu Amoda	1	$3.00
Winslow Homer	1	$3.00
Ekene Maduka	1	$3.00
Edward Hopper	1	$4.00
Georgia O'Keeffe	1	$4.00
Koki Tanaka	1	$4.00
Shakir Ali	1	$4.00
Nandalal Bose	1	$4.00
Tatsuo Miyajima	1	$4.00
Albert Bierstadt	1	$4.00
Li Chen	1	$4.00
Jasper Johns	1	$5.00
Raja Ravi Varma	1	$5.00
Modupeola Fadugba	1	$6.00
Zhan Wang	2	$8.00
Kalipada Ghoshal	2	$9.00
Sunil Das	3	$15.00
Abdur Rahman	3	$24.00
Manishi Dey	6	$81.00
Ismail Gulg	7	$95.00

Show all | Number of rows: 25 Filter rows: Search this table

Console

Q14. Execute a command to delete a record that violates a **referential integrity constraint.** State the message produced by the DBMS.

DELETE FROM customer WHERE cId = 2;

Error

SQL query: Copy

```
DELETE FROM customer WHERE cId = 2;
```

MySQL said:

```
#1451 - Cannot delete or update a parent row: a foreign key constraint fails (`uta_sprouts_2`.`store_customer`, CONSTRAINT `FK_STORE_CUSTOMERID` FOREIGN KEY (`cId`)
REFERENCES `customer` (`cId`))
```

Q15 Execute 3 insert commands for any table that attempt to insert records, such that the records violate the **explicit schema-based constraints (Key, Entity Integrity, Referential Integrity constraints)**. Make each of the 3 records violate a different types of integrity constraint. Include the insert statements and the error messages produced.

Key Constraint violation

INSERT INTO EMPLOYEE VALUES ('1', '123456781', 'Shubash Muniyappa', '10 Ray Street', 'Arlington', 'TX', '76019-1111', 'F', '1996-08-19', '2021-01-20', NULL, 'Associate 2', '30000', NULL, NULL, NULL, NULL);

Error

SQL query: Copy

```
INSERT INTO EMPLOYEE VALUES ('1', '123456781', 'Shubash Muniyappa', '10 Ray Street', 'Arlington', 'TX', '76019-1111', 'F', '1996-08-19', '2021-01-20', NULL, 'Associate 2', '30000', NULL, NULL, NU
```

MySQL said:

```
#1062 - Duplicate entry '123456781' for key 'PRIMARY'
```

Entity Integrity

INSERT INTO EMPLOYEE VALUES ('1', NULL , 'Shubash Muniyappa', '10 Ray Street', 'Arlington', 'TX', '76019-1111', 'F', '1996-08-19', '2021-01-20', NULL, 'Associate 2', '30000', NULL, NULL, NULL, NULL);

Error

SQL query: Copy

```
INSERT INTO EMPLOYEE VALUES ('1', NULL , 'Shubash Muniyappa', '10 Ray Street', 'Arlington', 'TX', '76019-1111', 'F', '1996-08-19', '2021-01-20', NULL, 'Associate 2', '30000', NULL, NULL, NULL, NU
```

MySQL said:

```
#1048 - Column 'SSN' cannot be null
```

Referential Integrity constraint violation

INSERT INTO `vendor_store` VALUES ('40', '20');

Error

SQL query: Copy

```
INSERT INTO `vendor_store` VALUES ('40', '20');
```

MySQL said:

```
#1452 - Cannot add or update a child row: a foreign key constraint fails (`uta_sprouts_2`.`vendor_store`, CONSTRAINT `PK_VENDOR_VENDORID` FOREIGN KEY (`vId`) REFERENCES
`vendor` (`vId`))
```

Q16 Execute an update command for store_customer table that attempts to update records in child rows, such that the record violate the foreign key constraint. Include the update statement and the error message.

UPDATE STORE_CUSTOMER SET sId = 2 WHERE cId = 1;

Error

SQL query: Copy

```
UPDATE STORE_CUSTOMER SET sId = 2 WHERE cId = 1;
```

MySQL said:

```
#1452 - Cannot add or update a child row: a foreign key constraint fails (`uta_sprouts_2`.`store_customer`, CONSTRAINT `FK_STORE_STOREID` FOREIGN KEY (`sId`) REFERENCES `store` (`sId`))
```

Contribution:

Both, Shubash Muniyappa and Sai Krishna Prateek contributed equally in each part of uploading and querying the database.

HONOR CODE

I Pledge, on my honor, to ~~uphe~~ uphold UTArlington's traditions of academic integrity, a tradition that values hard work & honest effort in the pursuit of academic excellence.

I Promise that I will submit only work that I Personally create or that I contribute to group collaborations & I will appropriately reference any work from other sources. I will follow the highest standards of integrity & ~~upho~~ uphold the spirit of the honor code.

I will not Participate in any form of Cheating/Sharing the questions/solutions.

SAI KRISHNA PRATEEK NAMA

UTAID :- 1001880903

Prateek

03/30/2023.

HONOR CODE

I pledge, on my honor, to uphold UT Arlington's tradition of academic integrity, a tradition that values hard work & honest effort in the pursuit of academic excellence.

I promise that I will submit only work that I personally create or that I contribute to group collaborations, and I will appropriately reference any work from other sources. I will follow the highest standards of integrity & uphold the spirit of the Honor Code.

I will not participate in any of cheating/sharing the questions/solutions.

SHUBASH MUNIYAPPA
1001915563
03/30/23