

Question 2

create table Warehouse (warehouseNumber int not null primary key, streetNumber int not null, streetName varchar(20) not null, postalCode char(6) not null, city varchar(20) not null)

Column name	Type schema	Type name	Length	Scale	Nulls
WAREHOUSENUMBER	SYSIBM	INTEGER	4	0	No
STREETNUMBER	SYSIBM	INTEGER	4	0	No
STREETNAME	SYSIBM	VARCHAR	20	0	No
POSTALCODE	SYSIBM	CHARACTER	6	0	No
CITY	SYSIBM	VARCHAR	20	0	No

create table Store (warehouseNumber int not null primary key, foreign key (warehouseNumber) references Warehouse, openingTime int not null, closingTime int not null)

Column name	Type schema	Type name	Length	Scale	Nulls
WAREHOUSENUMBER	SYSIBM	INTEGER	4	0	No
OPENINGTIME	SYSIBM	INTEGER	4	0	No
CLOSINGTIME	SYSIBM	INTEGER	4	0	No

create table Game (gameId int not null primary key, name varchar(30) not null, gameType varchar(15), developer varchar(30) not null, releaseDate date not null, averageRating int)

Column name	Type schema	Type name	Length	Scale	Nulls
GAMEID	SYSIBM	INTEGER	4	0	No
NAME	SYSIBM	VARCHAR	30	0	No
GAMETYPE	SYSIBM	VARCHAR	15	0	Yes
DEVELOPER	SYSIBM	VARCHAR	30	0	No
RELEASEDATE	SYSIBM	DATE	4	0	No
AVERAGERATING	SYSIBM	INTEGER	4	0	Yes

create table Console (consoleId int not null primary key, name varchar(30) not null, brand varchar(15) not null, averageRating int)

Column name	Type schema	Type name	Length	Scale	Nulls
CONSOLEID	SYSIBM	INTEGER	4	0	No
NAME	SYSIBM	VARCHAR	30	0	No
BRAND	SYSIBM	VARCHAR	15	0	No
AVERAGERATING	SYSIBM	INTEGER	4	0	Yes

create table Accessory (accessoryId int not null primary key, name varchar(30) not null, AccessoryType varchar(15) not null, brand varchar(15) not null, averageRating int)

Column name	Type schema	Type name	Length	Scale	Nulls
ACCESSORYID	SYSIBM	INTEGER	4	0	No
NAME	SYSIBM	VARCHAR	30	0	No
ACCESSORYTYPE	SYSIBM	VARCHAR	15	0	No

BRAND	SYSIBM	VARCHAR	15	0	No
AVERAGERATING	SYSIBM	INTEGER	4	0	Yes

create table ProductInstance (UPC bigint not null primary key, price int not null)

Column name	Type schema	Type name	Length	Scale	Nulls
UPC	SYSIBM	BIGINT	8	0	No
PRICE	SYSIBM	INTEGER	4	0	No

create table GameInstance(UPC bigint not null, foreign key (UPC) references ProductInstance, consoleId int not null, gameId int not null, primary key (consoleId, gameId, UPC), foreign key (gameId) references Game,foreign key (consoleId) references Console)

Column name	Type schema	Type name	Length	Scale	Nulls
UPC	SYSIBM	BIGINT	8	0	No
CONSOLEID	SYSIBM	INTEGER	4	0	No
GAMEID	SYSIBM	INTEGER	4	0	No

create table ConsoleInstance(UPC bigint not null, foreign key (UPC) references ProductInstance, consoleId int not null, primary key (consoleId, UPC), foreign key (consoleId) references Console, colour varchar(10), version varchar(25))

Column name	Type schema	Type name	Length	Scale	Nulls
UPC	SYSIBM	BIGINT	8	0	No
CONSOLEID	SYSIBM	INTEGER	4	0	No
COLOUR	SYSIBM	VARCHAR	10	0	Yes
VERSION	SYSIBM	VARCHAR	25	0	Yes

create table AccessoryInstance(UPC bigint not null, foreign key (UPC) references ProductInstance, accessoryId int not null, primary key (accessoryId, UPC), foreign key (accessoryId) references Accessory, colour varchar(10))

Column name	Type schema	Type name	Length	Scale	Nulls
UPC	SYSIBM	BIGINT	8	0	No
ACCESSORYID	SYSIBM	INTEGER	4	0	No
COLOUR	SYSIBM	VARCHAR	10	0	Yes

create table AccountHolder (username varchar(40) not null primary key, name varchar(30) not null, creditCardNumber bigint not null, billingStreetNumber int not null, billingStreetName varchar(20) not null,

billingPostalCode char(6) not null)

Column name	Type schema	Type name	Length	Scale	Nulls
USERNAME	SYSIBM	VARCHAR	40	0	No
NAME	SYSIBM	VARCHAR	30	0	No
CREDITCARDNUMBER	SYSIBM	BIGINT	8	0	No
BILLINGSTREETNUMBER	SYSIBM	INTEGER	4	0	No
BILLINGSTREETNAME	SYSIBM	VARCHAR	20	0	No
BILLINGPOSTALCODE	SYSIBM	CHARACTER	6	0	No

create table OrderInfo (orderNumber int not null primary key, username varchar(40), foreign key (username) references AccountHolder, shippingStreetNumber int not null, shippingStreetName varchar(20) not null, shippingPostalCode char(6) not null, orderDate date)

Column name	Type schema	Type name	Length	Scale	Nulls
ORDERNUMBER	SYSIBM	INTEGER	4	0	No
USERNAME	SYSIBM	VARCHAR	40	0	Yes
SHIPPINGSTREETNUMBER	SYSIBM	INTEGER	4	0	No
SHIPPINGSTREETNAME	SYSIBM	VARCHAR	20	0	No
SHIPPINGPOSTALCODE	SYSIBM	CHARACTER	6	0	No
ORDERDATE	SYSIBM	DATE	4	0	Yes

create table Stock (warehouseNumber int not null, foreign key (warehouseNumber) references Warehouse, UPC bigint not null, foreign key (UPC) references ProductInstance, quantity int not null)

Column name	Type schema	Type name	Length	Scale	Nulls
WAREHOUSENUMBER	SYSIBM	INTEGER	4	0	No
UPC	SYSIBM	BIGINT	8	0	No
QUANTITY	SYSIBM	INTEGER	4	0	No

create table Ordered (orderNumber int not null, foreign key (orderNumber) references OrderInfo, UPC bigint not null, foreign key (UPC) references ProductInstance, primary key (orderNumber, UPC), quantity int not null)

Column name	Type schema	Type name	Length	Scale	Nulls
ORDERNUMBER	SYSIBM	INTEGER	4	0	No
UPC	SYSIBM	BIGINT	8	0	No
QUANTITY	SYSIBM	INTEGER	4	0	No

create table Rating (username varchar(40) not null, foreign key (username) references AccountHolder, UPC bigint not null, foreign key (UPC) references ProductInstance, rating int not null, primary key (username, UPC, rating))

Column name	Type schema	Type name	Length	Scale	Nulls
USERNAME	SYSIBM	VARCHAR	40	0	No

UPC	SYSIBM	BIGINT	8	0	No
RATING	SYSIBM	INTEGER	4	0	No

Question 3

Below, I include a short list of only 5 Insert statements to demonstrate the results.

```
db2 => INSERT INTO AccountHolder VALUES('xoxox_mandy42_xoxox', 'Amanda Dajiero',
4729371537593726, 876, 'Tupper Street', 'H2A9J3')
DB20000I The SQL command completed successfully.
```

```
SELECT * FROM AccountHolder
USERNAME                                NAME
CREDITCARDNUMBER    BILLINGSTREETNUMBER BILLINGSTREETNAME    BILLINGPOSTALCODE
-----
xoxox_mandy42_xoxox                                Amanda Dajiero
4729371537593726                                876 Tupper Street    H2A9J3
```

```
db2 => INSERT INTO Rating VALUES ('xoxox_mandy42_xoxox', 22222223333, 8)
DB20000I The SQL command completed successfully.
```

```
SELECT * FROM AccountHolder
USERNAME                                NAME
CREDITCARDNUMBER    BILLINGSTREETNUMBER BILLINGSTREETNAME    BILLINGPOSTALCODE
-----
xoxox_mandy42_xoxox                                Amanda Dajiero
4729371537593726                                876 Tupper Street    H2A9J3
```

```
db2 => INSERT INTO Stock VALUES(8, 276776666777, 4)
DB20000I The SQL command completed successfully.
```

```
SELECT * FROM Stock
WAREHOUSENUMBER UPC                                QUANTITY
-----
8                276776666777                    4
```

```
db2 => INSERT INTO OrderInfo VALUES (16382917, 'xoxox_mandy42_xoxox', 876, 'Tupper
Street', 'H2A9J3', '2012-01-14')
DB20000I The SQL command completed successfully.
```

```
SELECT * FROM OrderInfo
ORDERNUMBER USERNAME                                SHIPPINGSTREETNUMBER
SHIPPINGSTREETNAME    SHIPPINGPOSTALCODE ORDERDATE
-----
16382917 xoxox_mandy42_xoxox                                876 Tupper
Street    H2A9J3    01/14/2012
```

```
db2 => INSERT INTO Ordered VALUES (16382917, 536984563217, 2)
```

DB20000I The SQL command completed successfully.

```
SELECT * FROM Ordered
ORDERNUMBER UPC                QUANTITY
-----
16382917      536984563217      2
```

Question 4

Below, I include a short list of only 5 tuples per table to demonstrate the results.

SELECT * FROM Warehouse FETCH FIRST 5 ROWS ONLY

WAREHOUSENUMBER	STREETNUMBER	STREETNAME	POSTALCODE	CITY
1	12	Rue Quebec	H7R5G3	Montreal
2	847	Ontario Avenue	M8D7J9	Toronto
3	9398	Alberta Terrace	T7R9I8	Calgary
4	1243	Rue Notre-Dame	K8D3W0	Québec
5	8722	Rang St-Antoine	L0G2Y6	Rimouski

SELECT * FROM Store FETCH FIRST 5 ROWS ONLY

WAREHOUSENUMBER	OPENINGTIME	CLOSINGTIME
1	10	21
2	9	21
4	10	17
6	12	17
10	10	17

SELECT * FROM Game FETCH FIRST 5 ROWS ONLY

GAMEID	NAME	GAMETYPE	DEVELOPER
RELEASEDATE	AVERAGERATING		
7856	Halo 4	FPS	Bungie
04/21/2012	-		
3847	Hello Kitty Island Adventure	Horror	EA
02/10/2010	-		
98	Super Smash Bros Melee	Fighting	Nintendo
01/07/2007	-		
13334	Conkers Bad Fur Day	Adventure	Tiger Studios
09/17/2008	-		
3256	Mario Party 47	Party	Nintendo
03/02/2014	-		

SELECT * FROM Console FETCH FIRST 5 ROWS ONLY

CONSOLEID	NAME	BRAND	AVERAGERATING
1	Xbox 360	Microsoft	-
4	Wii U	Nintendo	-
5	PS3	Sony	-
6	PS4	Sony	-
7	GameCube	Nintendo	-

SELECT * FROM Accessory FETCH FIRST 5 ROWS ONLY

ACCESSORYID	NAME	ACCESSORYTYPE	BRAND	AVERAGERATING
2	PS2 Controller	Control	Sony	0
3	PS3 Dual Shock Control	Control	Sony	0
4	Xbox 360 Control	Controller	Microsoft	0
5	Wii Control	Controller	Nintendo	0
6	Angry Potatoes Soft Toy	Soft Toy	Potato Studio	0

SELECT * FROM ProductInstance FETCH FIRST 5 ROWS ONLY

UPC	PRICE
546983012564	20
596348620987	50
785136987423	70
536984563217	60
615893517053	60

SELECT * FROM GameInstance FETCH FIRST 5 ROWS ONLY

UPC	CONSOLEID	GAMEID
123847877887	1	998
123432345678	1	7856
123847999991	2	43
123847811111	2	7856
123888881111	2	98237

SELECT * FROM ConsoleInstance FETCH FIRST 5 ROWS ONLY

UPC	CONSOLEID	COLOUR	VERSION
-----	-----------	--------	---------

938465728111	1 Black	250GB with Kinect
735495783831	1 Black	-
936677766661	1 White	-
108394804391	2 Black	-
888894894391	2 White	-

SELECT * FROM AccessoryInstance FETCH FIRST 5 ROWS ONLY

UPC	ACCESSORYID	COLOUR
546983012564	1	Black
596348620987	5	Pink
536984563217	7	Yellow
615893517053	8	Orange
789630125473	9	Green

SELECT * FROM AccountHolder FETCH FIRST 5 ROWS ONLY

USERNAME	NAME
CREDITCARDNUMBER	BILLINGSTREETNUMBER BILLINGSTREETNAME BILLINGPOSTALCODE
Stephen_harper	Bill Paxton
4517391740586618	8708 Parliament Road P2G5Z5
labrador_lover	Sarah Poodle
2957673957593723	988 Chihuahua Avenue S6J5C1
pink_mittens	Arnold Musclemann
2394758294720327	3216 Igloo Drive X2B4Z7
iLikeVideoGames	Leonardo Leonardo
1947682947253142	143 Nunavut Avenue F6A1H8
youremailadresshere	Seymour Butts
4860932657342323	18009 Evergreen Terrace L1X3A7

SELECT * FROM OrderInfo FETCH FIRST 5 ROWS ONLY

ORDERNUMBER	USERNAME	SHIPPINGSTREETNUMBER
SHIPPINGSTREETNAME	SHIPPINGPOSTALCODE	ORDERDATE
16382917	xoxox_mandy42_xoxox	876 Tupper
Street	H2A9J3	01/14/2012
45435658	Stephen_harper	8708 Parliament
Road	P2G5Z5	12/29/2010
92078798	pink_mittens	3216 Igloo Drive
X2B4Z7	02/28/2014	
90098908	iLikeVideoGames	143 Nunavut

Avenue	F6A1H8	01/01/2010	
65656758	iLikeVideoGames		143 Nunavut
Avenue	F6A1H8	01/01/2012	

SELECT * FROM Stock FETCH FIRST 5 ROWS ONLY

WAREHOUSENUMBER	UPC	QUANTITY
1	123432345678	10
1	123847877887	9
1	123847811111	14
1	123888881111	30
1	123847999991	23

SELECT * FROM Ordered FETCH FIRST 5 ROWS ONLY

ORDERNUMBER	UPC	QUANTITY
16382917	536984563217	2
45435658	599843784387	1
92078798	509678987277	1
92078798	111288888122	1
90098908	167777777744	1

SELECT * FROM Rating FETCH FIRST 5 ROWS ONLY

USERNAME	UPC	RATING
Stephen_harper	121232232331	10
Stephen_harper	123888881111	5
Stephen_harper	167777777744	1
Stephen_harper	565666555554	9
Stephen_harper	938465728111	7

Question 5

Query 1

This query returns the streetNumber, streetName and city of the stores having in stock at least one copy of the game 'Halo 4' for 'Xbox 360'.

SQL statement:

```
SELECT W.streetNumber, W.streetName, W.city
FROM store S, stock ST, warehouse W
WHERE ST.warehouseNumber = S.warehouseNumber
      AND W.warehouseNumber = ST.warehouseNumber
      AND ST.UPC = (SELECT GI.UPC
                    FROM Game G, GameInstance GI
                    WHERE G.name = 'Halo 4'
                      AND G.platform = 'Xbox 360')
```

```

WHERE G.name = 'Halo 4'
AND G.gameId = GI.gameId
AND GI.consoleId = (SELECT C.consoleId
                    FROM Console C
                    WHERE C.name = 'Xbox 360'))

AND ST.quantity >0

```

```

db2.cs.mcgill.ca - PuTTY
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 => SELECT W.streetNumber, W.streetName, W.city FROM store S, stock ST, warehouse W WHERE ST.warehouseNumber
= S.warehouseNumber AND W.warehouseNumber = ST.warehouseNumber AND ST.UPC = (SELECT GI.UPC FROM Game G, GameIn
stance GI WHERE G.name = 'Halo 4' AND G.gameId = GI.gameId AND GI.consoleId = (SELECT C.consoleId FROM Console
C WHERE C.name = 'Xbox 360')) AND ST.quantity >0

STREETNUMBER STREETNAME CITY
-----
12 Rue Quebec Montreal
847 Ontario Avenue Toronto
1243 Rue Notre-Dame Québec
2222 Stooges Road Regina

4 record(s) selected.

db2 =>

```

Execution:

The part

```

SELECT GI.UPC
FROM Game G, GameInstance GI
WHERE G.name = 'Halo 4'
AND G.gameId = GI.gameId
AND GI.consoleId = (SELECT C.consoleId FROM Console C WHERE C.name = 'Xbox 360')

```

gives UPC of the gameInstance 'Halo 4' on the console 'Xbox 360' which is 123432345678

```

SELECT name, gameId FROM game WHERE name = 'Halo 4'

```

NAME	GAMEID
Halo 4	7856

```

SELECT name, consoleId FROM Console WHERE name = 'Xbox 360'

```

NAME	CONSOLEID
Xbox 360	1

```

SELECT * FROM gameinstance WHERE gameId = 7856 AND consoleId = 1

```

UPC	CONSOLEID	GAMEID
123432345678	1	7856

Using

SELECT warehouseNumber FROM stock WHERE UPC = 123432345678 AND quantity >0
we get the list of all the warehouses having in stock this UPC:

WAREHOUSENUMBER

```
-----
1
2
4
5
7
8
9
11
12
```

SELECT warehouseNumber FROM store
gives the list of all the stores

WAREHOUSENUMBER

```
-----
1
2
3
4
6
10
12
```

so only the warehouses with warehouseNumber 1,2,4 and 12 are also stores
Using

```
SELECT * FROM warehouse
WHERE warehouseNumber = 1
OR warehouseNumber = 2
OR warehouseNumber = 4
OR warehouseNumber = 12
```

we can see:

WAREHOUSENUMBER	STREETNUMBER	STREETNAME	POSTALCODE	CITY
1	12	Rue Quebec	H7R5G3	Montreal
2	847	Ontario Avenue	M8D7J9	Toronto
4	1243	Rue Notre-Dame	K8D3W0	Québec
12	2222	Stooges Road	P0H6D3	Regina

which is the same as the result bellow.

Result of the query:

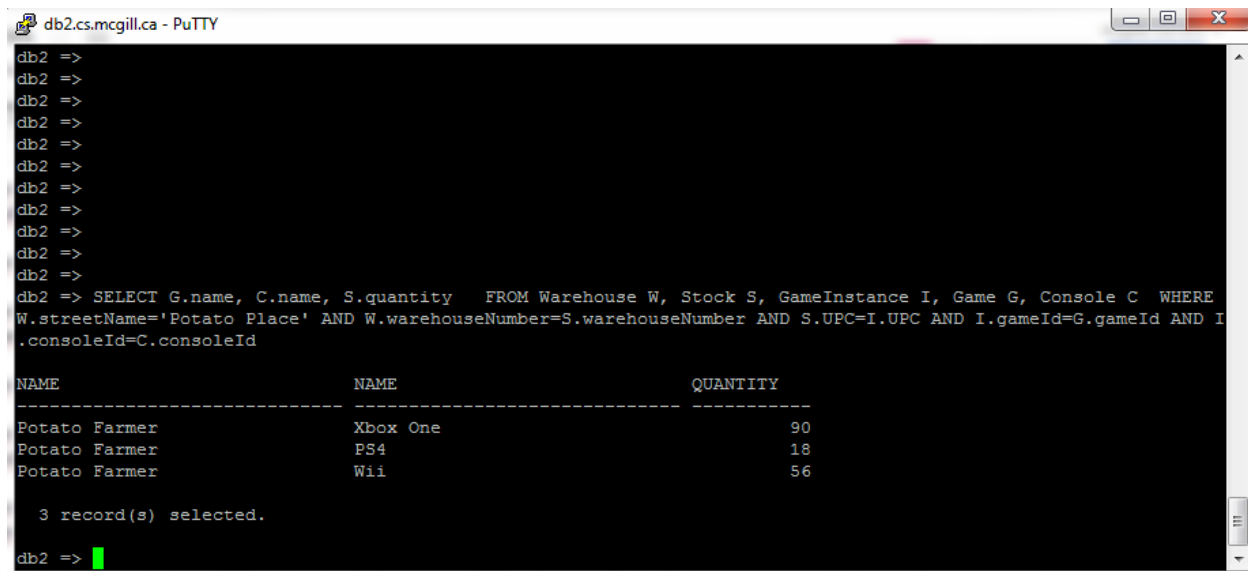
STREETNUMBER	STREETNAME	CITY
12	Rue Quebec	Montreal
847	Ontario Avenue	Toronto
1243	Rue Notre-Dame	Québec

Query 2

This query returns a list of the names of the Games which can be found at the Potato Place location as well as the console they can be played on and the quantity in stock.

SQL Statement:

```
SELECT G.name, C.name, S.quantity
FROM Warehouse W, Stock S, GameInstance I, Game G, Console C
WHERE W.streetName = 'Potato Place'
      AND W.warehouseNumber = S.warehouseNumber
      AND S.UPC = I.UPC
      AND I.gameId = G.gameId
      AND I.consoleId = C.consoleId
```



```
db2.cs.mcgill.ca - PuTTY
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 => SELECT G.name, C.name, S.quantity FROM Warehouse W, Stock S, GameInstance I, Game G, Console C WHERE
W.streetName='Potato Place' AND W.warehouseNumber=S.warehouseNumber AND S.UPC=I.UPC AND I.gameId=G.gameId AND I
.consoleId=C.consoleId

NAME                NAME                QUANTITY
-----
Potato Farmer       Xbox One              90
Potato Farmer       PS4                   18
Potato Farmer       Wii                   56

3 record(s) selected.

db2 => █
```

Execution:

In order to filter out the unwanted elements of each table, we use the sequential AND statements:

`W.streetName = 'Potato Place' AND W.warehouseNumber = S.warehouseNumber`

allows us to narrow the tuples down to only get the items in stock at the specific Warehouse.

We then use the statement `S.UPC = I.UPC` to fetch the UPCs of the corresponding ProductInstances in the GameInstance table and then use `I.gameId = G.gameId AND I.consoleId = C.consoleId` to map the GameInstance info

to a game and console respectively in order to get their names.

Relations Used:

Warehouse

Column name	Type schema	Type name	Length	Scale	Nulls
WAREHOUSENUMBER	SYSIBM	INTEGER	4	0	No
STREETNUMBER	SYSIBM	INTEGER	4	0	No
STREETNAME	SYSIBM	VARCHAR	20	0	No
POSTALCODE	SYSIBM	CHARACTER	6	0	No
CITY	SYSIBM	VARCHAR	20	0	No

Stock

Column name	Type schema	Type name	Length	Scale	Nulls
WAREHOUSENUMBER	SYSIBM	INTEGER	4	0	No
UPC	SYSIBM	BIGINT	8	0	No
QUANTITY	SYSIBM	INTEGER	4	0	No

GameInstance

Column name	Type schema	Type name	Length	Scale	Nulls
UPC	SYSIBM	BIGINT	8	0	No
CONSOLEID	SYSIBM	INTEGER	4	0	No
GAMEID	SYSIBM	INTEGER	4	0	No

Game

Column name	Type schema	Type name	Length	Scale	Nulls
GAMEID	SYSIBM	INTEGER	4	0	No
NAME	SYSIBM	VARCHAR	30	0	No
gametype	SYSIBM	VARCHAR	15	0	Yes
DEVELOPER	SYSIBM	VARCHAR	30	0	No
RELEASEDATE	SYSIBM	DATE	4	0	No
AVERAGERATING	SYSIBM	INTEGER	4	0	Yes

Console

Column name	Type schema	Type name	Length	Scale	Nulls
CONSOLEID	SYSIBM	INTEGER	4	0	No
NAME	SYSIBM	VARCHAR	30	0	No
BRAND	SYSIBM	VARCHAR	15	0	No
AVERAGERATING	SYSIBM	INTEGER	4	0	Yes

Query 3

This query provides the name of any AccountHolders who bought a Wii.

SQL Statement:

SELECT F.userName

```

FROM Console C, ConsoleInstance I, Ordered O, OrderInfo F
WHERE C.name = 'Wii'
      AND C.consoleId=I.consoleId
      AND I.UPC=O.UPC
      AND O.ordernumber=F.ordernumber

```

```

db2.cs.mcgill.ca - PuTTY
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 =>
db2 => SELECT F.userName FROM Console C, ConsoleInstance I, Ordered O, OrderInfo F WHERE C.name ='Wii' AND C.c
onsoleId=I.consoleId AND I.UPC=O.UPC AND O.ordernumber=F.ordernumber

USERNAME
-----
super_hunk

1 record(s) selected.
db2 =>

```

Execution

We used the sequential “AND” statements in the “WHERE” clause to gradually limit the tuples of the combination of tables.

We start by searching for the consoleId of the Wii using `C.name = 'Wii'`.

Then, we use that consoleId to get all corresponding UPC (meaning we include all colors):
`C.consoleId=I.consoleId`.

From that point, we use the UPC to find the console in the Ordered (Products) list using `I.UPC=O.UPC`.

Finally, we use `O.ordernumber=F.ordernumber` to find the Orders where the Wii was purchased and thus find the usernames.

Relations Used

Console

Column	Type	Type	Length	Scale	Nulls
name	schema	name			
-----	-----	-----	-----	-----	-----
CONSOLEID	SYSIBM	INTEGER	4	0	No
NAME	SYSIBM	VARCHAR	30	0	No
BRAND	SYSIBM	VARCHAR	15	0	No
AVERAGERATING	SYSIBM	INTEGER	4	0	Yes

ConsoleInstance

Column name	Type schema	Type name	Length	Scale	Nulls
UPC	SYSIBM	BIGINT	8	0	No
CONSOLEID	SYSIBM	INTEGER	4	0	No
COLOUR	SYSIBM	VARCHAR	10	0	Yes
VERSION	SYSIBM	VARCHAR	25	0	Yes

Ordered

Column name	Type schema	Type name	Length	Scale	Nulls
ORDERNUMBER	SYSIBM	INTEGER	4	0	No
UPC	SYSIBM	BIGINT	8	0	No
QUANTITY	SYSIBM	INTEGER	4	0	No

OrderInfo

Column name	Type schema	Type name	Length	Scale	Nulls
ORDERNUMBER	SYSIBM	INTEGER	4	0	No
USERNAME	SYSIBM	VARCHAR	40	0	Yes
SHIPPINGSTREETNUMBER	SYSIBM	INTEGER	4	0	No
SHIPPINGSTREETNAME	SYSIBM	VARCHAR	20	0	No
SHIPPINGPOSTALCODE	SYSIBM	CHARACTER	6	0	No
ORDERDATE	SYSIBM	DATE	4	0	Yes

Query 4

This query will return all the stock from different warehouse with average rating of a specify product via their UPC .

SQL Statement

UPC without rating

```

SELECT S.UPC, SUM(S.quantity) AS Quantity , AVG(R.rating) AS Rating
FROM stock S
LEFT OUTER JOIN rating R
ON S.UPC = R.UPC
WHERE 615893517053= S.UPC
GROUP by S.UPC

```

```
db2 =>
SELECT S.UPC, SUM(S.quantity) AS Quantity , AVG(R.rating) AS Rating FROM stock S LEFT OUTER JOIN rating R ON
S.UPC = R.UPC WHERE 615893517053= S.UPC GROUP by S.UPC

db2 =>
UPC                QUANTITY    RATING
-----
615893517053      340          -

1 record(s) selected.

db2 => db2 => █
```

Connected to db2.cs.mcgill.ca SSH2 - aes128-cbc - hmac-md5 - n 109x47 NUM

UPC with rating

```
SELECT S.UPC, SUM(S.quantity) AS Quantity , AVG(R.rating) AS Rating
FROM stock S
LEFT OUTER JOIN rating R
ON S.UPC = R.UPC
WHERE 108394804391 = S.UPC
GROUP by S.UPC
```

```
db2 =>
SELECT S.UPC, SUM(S.quantity) AS Quantity , AVG(R.rating) AS Rating FROM stock S LEFT OUTER JOIN rating R ON
S.UPC = R.UPC WHERE 108394804391 = S.UPC GROUP by S.UPC

db2 =>
UPC                QUANTITY    RATING
-----
108394804391      122          6

1 record(s) selected.

db2 => db2 => █
```

Execution

We used the sequential “LEFT OUTER JOIN” statements to join same UPC which include all Stock UPC and UPCs that both Rating and Stock have. In this case, even when rating is empty for some UPCs, we will still be able to display corresponding UPCs and its remaining quantities.

The WHERE statement make sure that we only display the required UPC.

Since a single UPC might have multiple rating we take the average of it and group it by it’s UPC number so there will no be more than one row return.

Relations Used

Stock

Column name	Type schema	Type name	Length	Scale	Nulls
WAREHOUSENUMBER	SYSIBM	INTEGER	4	0	No
UPC	SYSIBM	BIGINT	8	0	No
QUANTITY	SYSIBM	INTEGER	4	0	No

Rating

Column name	Type schema	Type name	Length	Scale	Nulls
USERNAME	SYSIBM	VARCHAR	40	0	No

UPC	SYSIBM	BIGINT	8	0	No
RATING	SYSIBM	INTEGER	4	0	No

Query 5

Show all the account holder that have string 'e' within its username and have order before

```
SELECT A.name, COUNT(O.ordernumber) AS #orders
FROM accountholder A, orderinfo O
WHERE A.username = O.username
AND A.username LIKE '%e%'
GROUP BY A.name
```

```
db2 => SELECT A.name, COUNT(O.ordernumber) AS #orders FROM accountholder A, orderinfo O WHERE A.username = O.username AND A.username LIKE '%e%' GROUP BY A.name
```

NAME	#ORDERS
Arnold Musclemann	1
Bill Paxton	1
Harry Pisino	1
Helga Strong	1
Leonardo Leonardo	3
Seymour Butts	1
Sue Vlaki	1

```
7 record(s) selected.
db2 =>
```

Execution

At the where statement limited to tuples where account holder username equal to username that use in order, hence only username that have used for order will be selected. Then use wildcard to select user name that contain string 'e' within it, if not, it will not be selected. Then group by it's username and display selected name with number of orders they made in the past.

Relations Used

OrderInfo

Column	Type	Type	Length	Scale	Nulls
name	schema	name			
ORDERNUMBER	SYSIBM	INTEGER	4	0	No
USERNAME	SYSIBM	VARCHAR	40	0	Yes
SHIPPINGSTREETNUMBER	SYSIBM	INTEGER	4	0	No
SHIPPINGSTREETNAME	SYSIBM	VARCHAR	20	0	No
SHIPPINGPOSTALCODE	SYSIBM	CHARACTER	6	0	No
ORDERDATE	SYSIBM	DATE	4	0	Yes

AccountHolder

Column	Type	Type	Length	Scale	Nulls
name	schema	name			
USERNAME	SYSIBM	VARCHAR	40	0	No
NAME	SYSIBM	VARCHAR	30	0	No
CREDITCARDNUMBER	SYSIBM	BIGINT	8	0	No
BILLINGSTREETNUMBER	SYSIBM	INTEGER	4	0	No
BILLINGSTREETNAME	SYSIBM	VARCHAR	20	0	No
BILLINGPOSTALCODE	SYSIBM	CHARACTER	6	0	No

Question 6

Data Modification 1

Delete all orders made before 2009-12-20 from the database

```
DELETE FROM Ordered O
WHERE O.orderNumber IN
      (SELECT OI.orderNumber
       FROM OrderInfo OI
       WHERE OI.orderDate < '2009-12-20' )
```

```
DELETE FROM OrderInfo
WHERE orderDate < '2009-12-20'
```

Before the deletes:

```
SELECT * FROM OrderInfo
```

ORDERNUMBER	USERNAME	SHIPPINGSTREETNUMBER	SHIPPINGSTREETNAME	SHIPPINGPOSTALCODE	ORDERDATE
34759827	Stephen_harper	8708	Parliament Road	P2G5Z5	02/09/2007
45435658	Stephen_harper	8708	Parliament Road	P2G5Z5	12/29/2010
23143214	labrador_lover	988	Chihuahua Avenue	S6J5C1	12/15/2009
92078798	pink_mittens	3216	Igloo Drive	X2B4Z7	02/28/2014
78265821	iLikeVideoGames	143	Nunavut Avenue	F6A1H8	02/09/2007
90098908	iLikeVideoGames	143	Nunavut Avenue	F6A1H8	01/01/2010
65656758	iLikeVideoGames	143	Nunavut Avenue	F6A1H8	01/01/2012
65857657	iLikeVideoGames	143	Nunavut Avenue	F6A1H8	01/01/2014
98987654	billy_bob_higgins	1231	Potato Road	A1G6W4	02/01/2014
15482639	billy_bob_higgins	1232	Spud Road	A1G6W4	02/08/2014
2936451	billy_bob_higgins	1233	Frie Road	A1G6W4	03/01/2014
74839348	trucks_are_cool	98977	Foreman Street	P2F5A2	02/09/2009
55453534	trucks_are_cool	98977	Foreman Street	P2F5A2	12/26/2010
11111111	silk_scarf	2323	Spice Road	P3V6B1	11/11/2012
87767334	jm_timmys	11111	Avenue Seigneurie	H1A5B8	02/28/2014
98776165	greek_food_lover	988665	Opa Avenue	A1C4S2	01/17/2014
9898872	super_hunk	545475	Beetroot Place	Y7A1C4	07/11/2013
16291603	youremailaddresshere	18009	Evergreen Terrace	L1X3A7	07/11/2013

```
SELECT * FROM Ordered
```

ORDERNUMBER	UPC	QUANTITY
34759827	121232232331	3
34759827	276776666777	1
34759827	167777777754	1
45435658	599843784387	1
23143214	122222233311	2
23143214	167744444444	1
92078798	509678987277	1
92078798	111288888122	1

78265821	12222233311	1
78265821	1767676767	1
78265821	27677666777	1
78265821	590998908987	1
90098908	16777777744	1
90098908	599843784387	1
90098908	16777777754	1
65656758	123987379999	1
65656758	111288888122	1
65656758	111222788122	1
16291603	2323232322	2
98987654	123888881111	1
98987654	108394804391	1
15482639	123888881111	1
2936451	123888881111	1
74839348	565666555554	1
55453534	123321123321	1
11111111	27677666777	5
87767334	123432345678	1
98776165	123333232333	1
9898872	1767676767	1
9898872	93999999911	1

After the deletes:

SELECT * FROM OrderInfo

ORDERNUMBER	USERNAME	SHIPPINGSTREETNUMBER	SHIPPINGSTREETNAME	SHIPPINGPOSTALCODE	ORDERDATE
45435658	Stephen_harper	8708	Parliament Road	P2G5Z5	12/29/2010
92078798	pink_mittens	3216	Igloo Drive	X2B4Z7	02/28/2014
90098908	iLikeVideoGames	143	Nunavut Avenue	F6A1H8	01/01/2010
65656758	iLikeVideoGames	143	Nunavut Avenue	F6A1H8	01/01/2012
65857657	iLikeVideoGames	143	Nunavut Avenue	F6A1H8	01/01/2014
98987654	billy_bob_higgins	1231	Potato Road	A1G6W4	02/01/2014
15482639	billy_bob_higgins	1232	Spud Road	A1G6W4	02/08/2014
2936451	billy_bob_higgins	1233	Frie Road	A1G6W4	03/01/2014
55453534	trucks_are_cool	98977	Foreman Street	P2F5A2	12/26/2010
11111111	silk_scarf	2323	Spice Road	P3V6B1	11/11/2012
87767334	jm_timmys	11111	Avenue Seigneurie	H1A5B8	02/28/2014
98776165	greek_food_lover	988665	Opa Avenue	A1C4S2	01/17/2014
9898872	super_hunk	545475	Beetroot Place	Y7A1C4	07/11/2013
16291603	youremailaddresshere	18009	Evergreen Terrace	L1X3A7	07/11/2013

ORDERNUMBER	UPC	QUANTITY
45435658	599843784387	1
92078798	509678987277	1
92078798	111288888122	1
90098908	167777777744	1
90098908	599843784387	1

90098908	16777777754	1
65656758	123987379999	1
65656758	111288888122	1
65656758	111222788122	1
16291603	2323232322	2
98987654	123888881111	1
98987654	108394804391	1
15482639	123888881111	1
2936451	123888881111	1
55453534	123321123321	1
11111111	276776666777	5
87767334	123432345678	1
98776165	123333232333	1
9898872	176767676767	1
9898872	93999999911	1

Data Modification 2

Add 10 copies of the game 'Mario Party 47' to the warehouses with warehouseNumber 1, 2 or 3

```
UPDATE Stock
SET quantity = quantity +10
WHERE UPC = (SELECT GI.UPC
              FROM Game G, GameInstance GI
              WHERE G.name = 'Mario Party 47'
              AND G.gameId = GI.gameId )
AND (warehouseNumber = 1
     OR warehouseNumber = 2
     OR warehouseNumber = 3)
```

Before the update:

```
SELECT * FROM Stock WHERE UPC = (SELECT GI.UPC FROM Game G, GameInstance GI WHERE G.name =
'Mario Party 47' AND G.gameId = GI.gameId )AND (warehouseNumber = 1 OR warehouseNumber = 2 OR
warehouseNumber = 3)
```

WAREHOUSENUMBER	UPC	QUANTITY
1	176767676767	34
2	176767676767	4
3	176767676767	23

After the update:

WAREHOUSENUMBER	UPC	QUANTITY
1	176767676767	44
2	176767676767	14
3	176767676767	33

Data Modification 3

Updates the average rating of the "Potato Farmer" game on the "Xbox One" console.

```
UPDATE Game
```

```

SET averageRating = (
    SELECT avg(R.rating)
    FROM Rating R, GameInstance I, Game G, Console C
    WHERE R.UPC=I.UPC
        AND I.GameId =G.GameId
        AND I.consoleId=C.consoleId
        AND G.name='Potato Farmer'
        AND C.name='Xbox One')

```

WHERE name='Potato Farmer'

Before the update:

db2 => select * from game

GAMEID	NAME	GAMETYPE	DEVELOPER	RELEASEDATE
AVERAGERATING				
----	----	----	----	----
----	7856 Halo 4	FPS	Bungie	04/21/2012
-	3847 Hello Kitty Island Adventure	Horror	EA	02/10/2010
-	98 Super Smash Bros Melee	Fighting	Nintendo	01/07/2007
-	13334 Conkers Bad Fur Day	Adventure	Tiger Studios	09/17/2008
-	3256 Mario Party 47	Party	Nintendo	03/02/2014
-	18497 Spyro Last Flame	Adventure	Insomniac Games	05/22/2011
-	67587 Final Fantasy LVIII	RPG	SquareEnix	03/05/2014
-	43 Crash Bandicoot	Adventure	Naughty Dog	06/10/2009
-	998 Guitar Hero: Miley Cyrus	Music	RockStar	01/20/2013
-	876 Forza 7 5/8	Racing	Motor Games	09/28/2013
-	56767 Wario World	Platformer	Nintendo	05/09/2010
-	8974 Barbie Ballet Princess	Dance	Mattel	06/11/2013
-	98237 Potato Farmer	Action	Charlotte Studios	02/29/2012
-	54656 Tekken 404	Fighting	Dojo Studios	03/14/2012
-				

After the update:

db2 => select * from Game

GAMEID	NAME	GAMETYPE	DEVELOPER	RELEASEDATE
AVERAGERATING				
----	----	----	----	----
----	7856 Halo 4	FPS	Bungie	04/21/2012
-	3847 Hello Kitty Island Adventure	Horror	EA	02/10/2010
-	98 Super Smash Bros Melee	Fighting	Nintendo	01/07/2007
-	13334 Conkers Bad Fur Day	Adventure	Tiger Studios	09/17/2008
-	3256 Mario Party 47	Party	Nintendo	03/02/2014
-	18497 Spyro Last Flame	Adventure	Insomniac Games	05/22/2011
-	67587 Final Fantasy LVIII	RPG	SquareEnix	03/05/2014
-	43 Crash Bandicoot	Adventure	Naughty Dog	06/10/2009
-				

-	998 Guitar Hero: Miley Cyrus	Music	RockStar	01/20/2013
-	876 Forza 7 5/8	Racing	Motor Games	09/28/2013
-	56767 Wario World	Platformer	Nintendo	05/09/2010
-	8974 Barbie Ballet Princess	Dance	Mattel	06/11/2013
-	98237 Potato Farmer	Action	Charlotte Studios	02/29/2012
6	54656 Tekken 404	Fighting	Dojo Studios	03/14/2012
-				

Data Modification 4

Increase the first product price in UPC (min UPC)order which have the max rating by 1000 times

```
UPDATE productinstance
SET price = price *1000
WHERE UPC = (SELECT MIN(r.UPC)
             FROM rating r
             WHERE r.rating>=10)
```

AFTER

```
db2 => select * from productinstance
```

UPC	PRICE
546983012564	20
596348620987	50
785136987423	70
536984563217	60
615893517053	60
789630125473	80
456258753951	90
731964280853	10
463020105060	20
152535957545	30
938465728111	300
735495783831	300
936677766661	320
108394804391	500
888894894391	550
239873987771	270
939999999911	270
936363636661	270
593292373731	300
121232232331	350000
124213476461	279
593987932841	400
867487987971	50
123432345678	40
123847877887	45
123847811111	60
123888881111	69
123847999991	25
122222233311	19
176767676767	34
276776666777	34
590998908987	40
167777777744	49
599843784387	59
167777777754	59
123333232333	14
232323232322	39
167744444444	25
878787877777	78
509678987277	68
123321123321	70
222222233333	50
565666555554	45
123987379999	12
111288888122	27
111222788122	19

46 record(s) selected.

BEFORE

```
db2 => select * from productinstance
```

UPC	PRICE
546983012564	20
596348620987	50
785136987423	70
536984563217	60
615893517053	60
789630125473	80
456258753951	90
731964280853	10
463020105060	20
152535957545	30
938465728111	300
735495783831	300
936677766661	320
108394804391	500
888894894391	550
239873987771	270
939999999911	270
936363636661	270
593292373731	300
121232232331	350
124213476461	279
593987932841	400
867487987971	50
123432345678	40
123847877887	45
123847811111	60
123888881111	69
123847999991	25
122222233311	19
176767676767	34
276776666777	34
590998908987	40
167777777744	49
599843784387	59
167777777754	59
123333232333	14
232323232322	39
167744444444	25
878787877777	78
509678987277	68
123321123321	70
222222233333	50
565666555554	45
123987379999	12
111288888122	27
111222788122	19

46 record(s) selected.

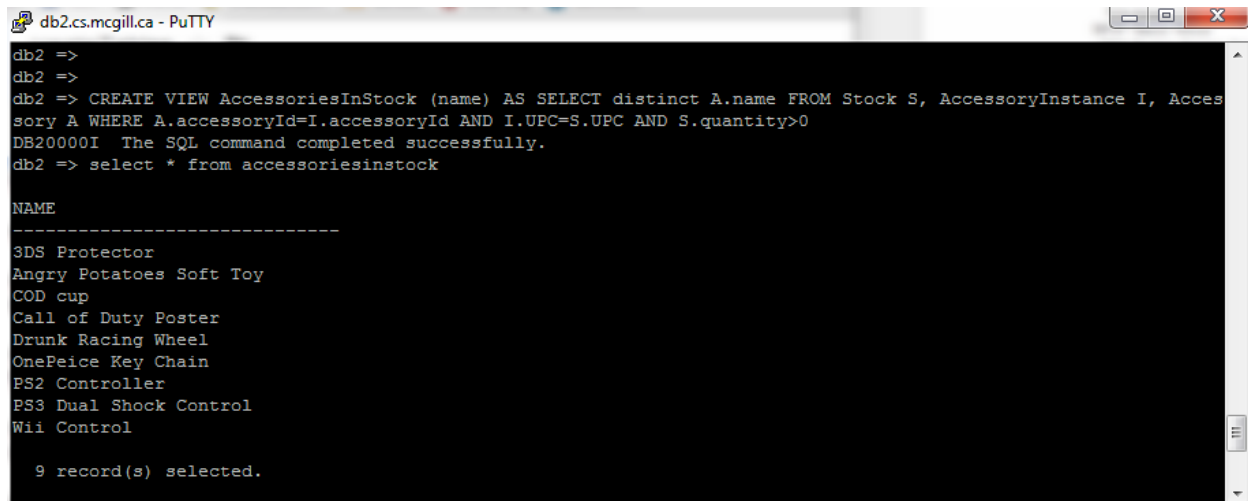
Question 7

View 1

```
CREATE VIEW AccessoriesInStock (name)
AS
    SELECT distinct A.name
    FROM Stock S, AccessoryInstance I, Accessory A
    WHERE A.accessoryId=I.accessoryId AND I.UPC=S.UPC AND S.quantity>0
```

```
CREATE VIEW AccessoriesInStock (name) AS SELECT distinct A.name FROM Stock S,
AccessoryInstance I, Accessory A WHERE A.accessoryId=I.accessoryId AND I.UPC=S.UPC AND
S.quantity>0
```

This simply represents all the Accessories that are currently in stock in some Warehouse.



```
db2.cs.mcgill.ca - PuTTY
db2 =>
db2 =>
db2 => CREATE VIEW AccessoriesInStock (name) AS SELECT distinct A.name FROM Stock S, AccessoryInstance I, Accessory A WHERE A.accessoryId=I.accessoryId AND I.UPC=S.UPC AND S.quantity>0
DB20000I The SQL command completed successfully.
db2 => select * from accessoriesinstock

NAME
-----
3DS Protector
Angry Potatoes Soft Toy
COD cup
Call of Duty Poster
Drunk Racing Wheel
OnePeice Key Chain
PS2 Controller
PS3 Dual Shock Control
Wii Control

9 record(s) selected.
```

View 2

```
CREATE VIEW ConsoleOver330(name) AS SELECT distinct x.name FROM productinstance P,
consoleinstance G, console x, console WHERE G.consoleid = x.consoleid AND P.UPC = G.UPC AND P.price
>=330
```

This simply represent all the Console that are priced over 330 dollars.

```
db2 => CREATE VIEW ConsoleOver330(name) AS SELECT distinct x.name FROM productinstance P, consoleinstance G, console x, console WHERE G.consoleid = x.consoleid AND P.UPC = G.UPC AND P.price >=330
DB20000I The SQL command completed successfully.
db2 => select * from ConsoleOver330

NAME
-----
Wii U
Xbox One

2 record(s) selected.
```

Question 8

Constraint 1

ALTER TABLE ProductInstance ADD CONSTRAINT priceConstraint CHECK (price > 0)
DB20000I The SQL command completed successfully.

Test the new constraint by inserting a tuple that violates it:

db2 => INSERT INTO ProductInstance VALUES(473847877337, -5)
DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned:
SQL0545N The requested operation is not allowed because a row does not satisfy the check constraint "CS421G42.PRODUCTINSTANCE.PRICECONSTRAINT".
SQLSTATE=23513

Constraint 2

ALTER TABLE Rating ADD CONSTRAINT ratingConstraint CHECK (rating >= 0 AND rating <=10)
DB20000I The SQL command completed successfully.

Test the new constraint by inserting a tuple that violates it:

db2 => INSERT INTO Rating VALUES ('Stephen_harper', 938465728111, 13)
DB21034E The command was processed as an SQL statement because it was not a valid Command Line Processor command. During SQL processing it returned:
SQL0545N The requested operation is not allowed because a row does not satisfy the check constraint "CS421G42.RATING.RATINGCONSTRAINT".
SQLSTATE=23513