Part 1: Conceptual Design - 40%

1. **Suggest a situation where you can use a database to manage and record daily transactions. (200-400 words).**

A brewing company sells beers to distributor or another retailer.
A brewing company has employee.
A distributor or retailer makes order by orderID from brewing company.
A beer could be sold by online store which is owned by brewing company.
One kind of beer only be produced in one company.
An order includes beers.

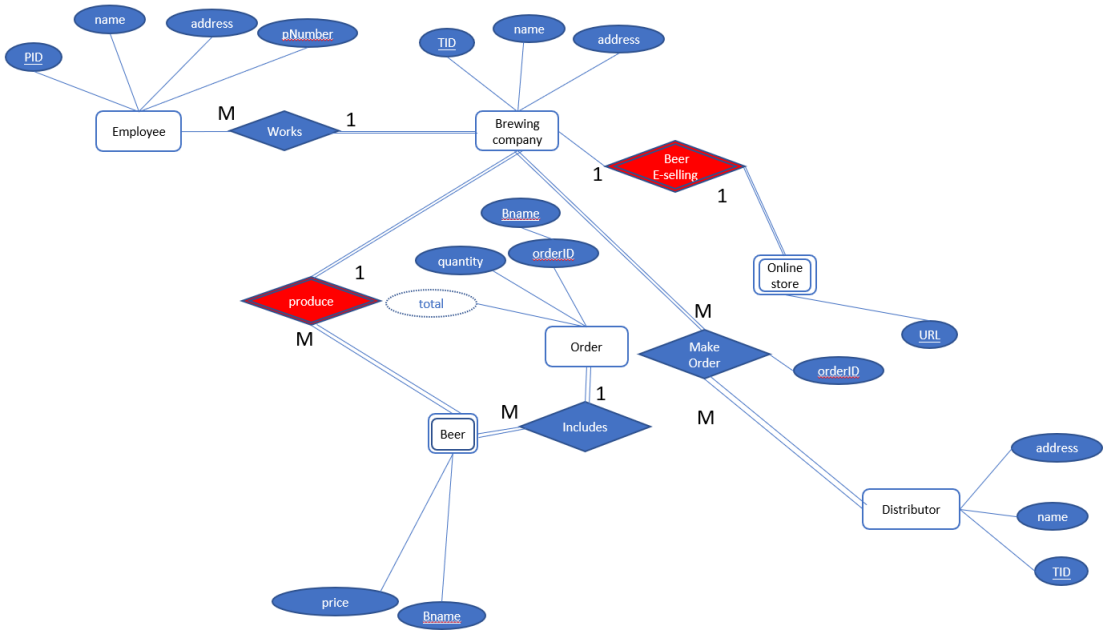
-A beer could beer name, price.
-A brew company could have TextID(BC_TID), address, name.
-A distributor or retailer could have TextID(D_TID), address, name.
-An employee could have Personal ID, address, name, phoneNumber.
-An online store could have an unique URL.
-An order could have orderID, beer name, quantity and total (a total could be derived by beer's price and quantity).
-Order's quantity times beer's price is total.
-An order contains same beer.

A brewing company could have coordination with several distributors or retailer, but just has one online-store on its own.
A distributor or retailer could have coordination with several brewing companies.
A brew company could have many employees, but an employee only works in one company.
A brewing company could have many kinds of beers, but one kind of beer is produced by one company.
An order could have many beers, but one beer only include in one order.

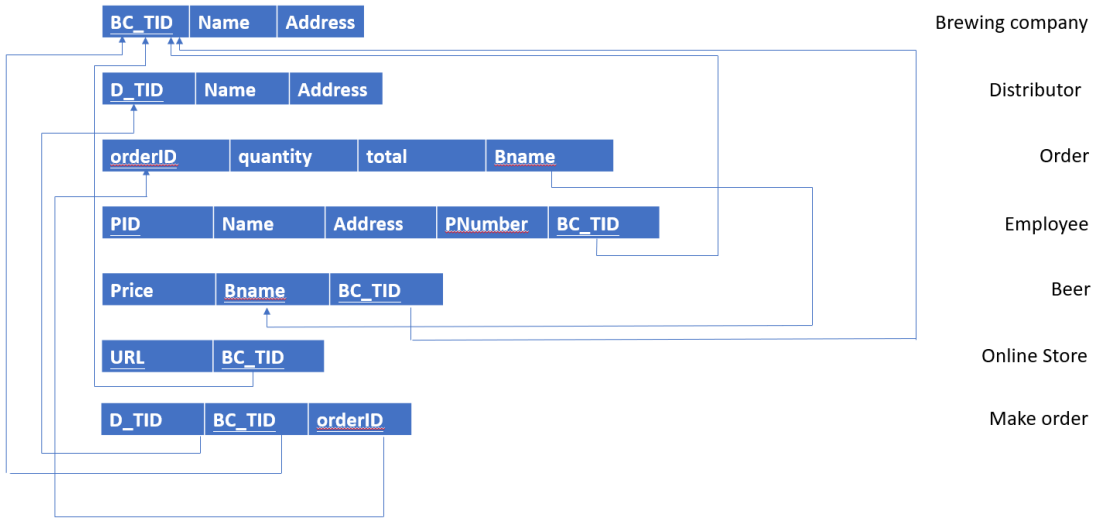
Not all brewing companies have their own online store.
Not all employees work on brewing companies.

2. **Draw a suitable ER diagram for your database; consider the following in your conceptual design:**

   - **Include at least five entities.**
   - **The relationship between entities should include one-to-one, one-to-many and many-to-many.**
   - **Provide an example of a derived attribute in your design.**



3. **Convert the conceptual design into a relational model.** ~~**Make sure that the tables are in a 3rd normal form.**~~



Part 2: Physical design - 20% (5 marks each)

1. Create the corresponding database using DDL

```
DROP DATABASE IF exists BREWING;
CREATE DATABASE BREWING;
USE BREWING;
```

2. Create all the necessary tables identified above using DDL

```
create table BC(
BC_TID int,
BC_name     varchar(40),
BC_address  varchar(40),
primary key (BC_TID)
);

create table BR(
Bname varchar(40),
price varchar(40),
BC_TID int ,
primary key (Bname),
foreign key (BC_TID)
    references BC (BC_TID)
);

create table OD(
OID varchar(40),
Bname varchar(40),
quantity varchar(40),
total varchar(40) ,
date date,
primary key (OID),
foreign key (Bname)
    references BR(Bname)
);
create table E(
PID      varchar(40),
name        varchar(40),
address  varchar(40),
PN  varchar(40),
BC_TID int,
primary key (PID),
foreign key (BC_TID)
    references BC(BC_TID)
);
```

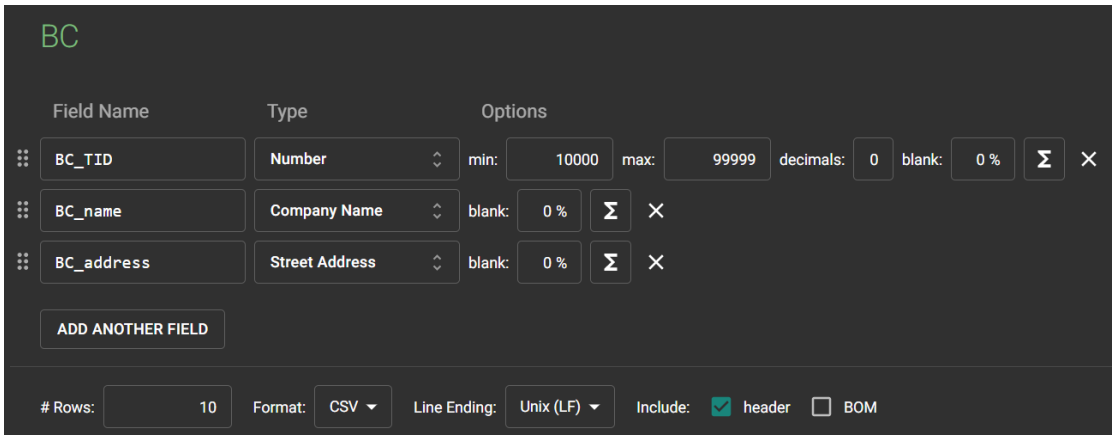
```sql
create table OS(
URL      varchar(40),
BC_TID      int,
primary key (URL),
foreign key (BC_TID)
    references BC(BC_TID)
);
create table D(
D_TID      varchar(40),
D_name      varchar(40),
D_address  varchar(40),
primary key (D_TID)
);
create table MO(
OID      varchar(40),
BC_TID      int,
D_TID varchar(40),

foreign key (OID)
    references OD(OID),
foreign key (BC_TID)
    references BC(BC_TID),
foreign key (D_TID)
    references D(D_TID)
);
```

3. Populate at least three of your tables with some data using DML (insert into statement)

```sql
insert into BC values ('5000','Hanne','Dulbin');
insert into E values ('sfs4df4e64a6ef64','Joe','01293 Oriole Parkway','085-123-4567','5000');
insert into OS values('http://www.google.com','5000');
```

4. Populate your database with a large data set representing a one-year transaction (01/01/2022 - 31/12/2022) on each table. (Use online data generators such as Mockaroo or generate data to generate synthetic data.)

Brewing Company



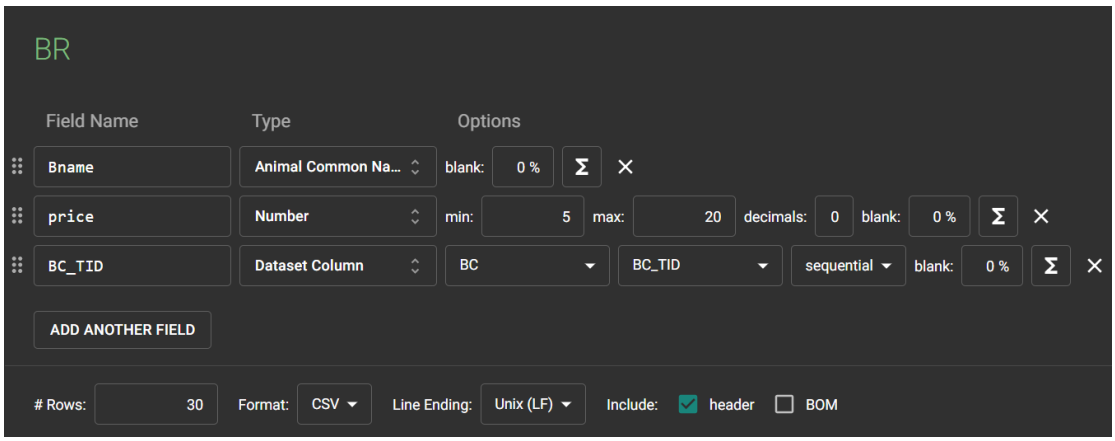
Data in Result Grid

```
73 •    select * from BC;
```

Result Grid | Filter Rows:

BC_TID	BC_name	BC_address
12275	Trilia	4 Vidon Point
14241	Avavee	27 Havey Street
29679	Vinder	289 Brickson Park Point
46970	Skipfire	5 Donald Court
47003	Tagcat	27 North Plaza
58077	Skilith	36 Prairie Rose Drive
70644	Browsedrive	53946 Lyons Place

Beer



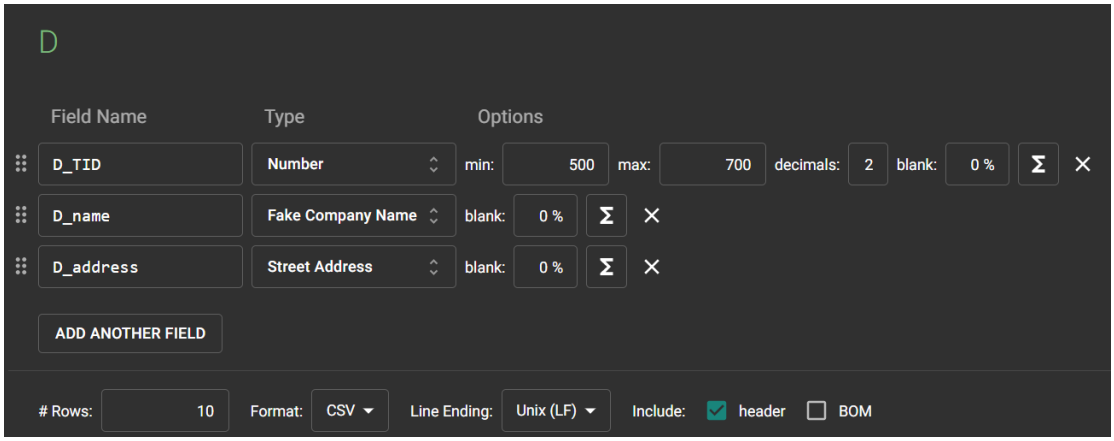
Data in Result Grid

```
78 •    select * from BR;
79
```

Result Grid | Filter Rows:

Bname	price	BC_TID
Australian brush turkey	15	47003
Bandicoot, long-nosed	11	95175
Barbet, levaillant's	18	58077
Boar, wild	18	46970
Bustard, stanley	9	47003
Chickadee, black-capped	12	29679
Covote	8	70644

Distributor



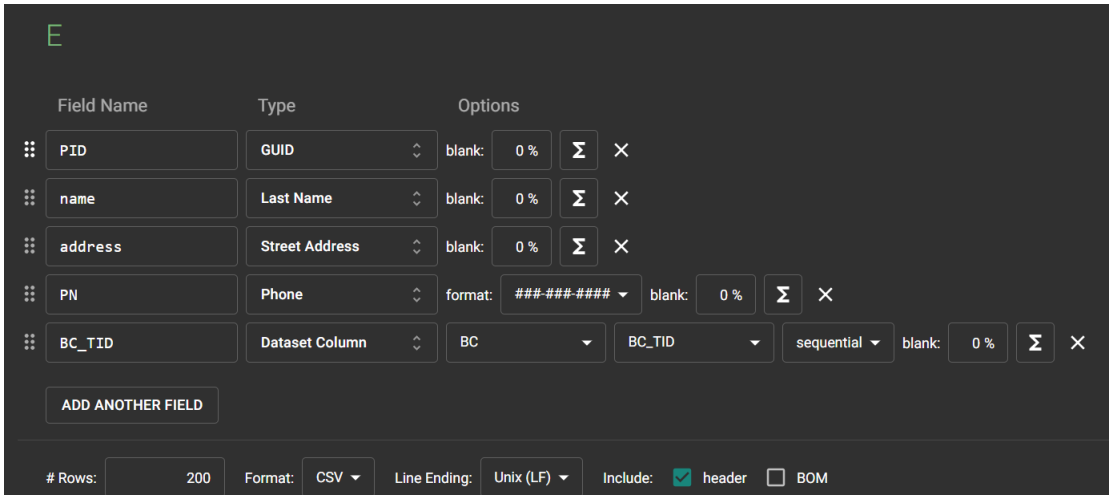
Data in Result Grid

```
74 •  select * from D;
75
```

Result Grid | Filter Rows: | Edit:

D_TID	D_name	D_address
545.51	Mitchell and Sons	27 Hazelcrest Avenue
562.15	Beatty Group	9653 Bunker Hill Crossing
574.12	Prosacco, Hoppe and Towne	7613 Straubel Center
575.29	Koepp-Monahan	63 Summerview Drive
584.24	Pfannerstill LLC	75 Meadow Valley Lane
594.84	Effertz, Pollich and Bins	216 Kedzie Street
608.13	McKenzie, Schaden and Williamson	4109 Merrick Circle

Employee



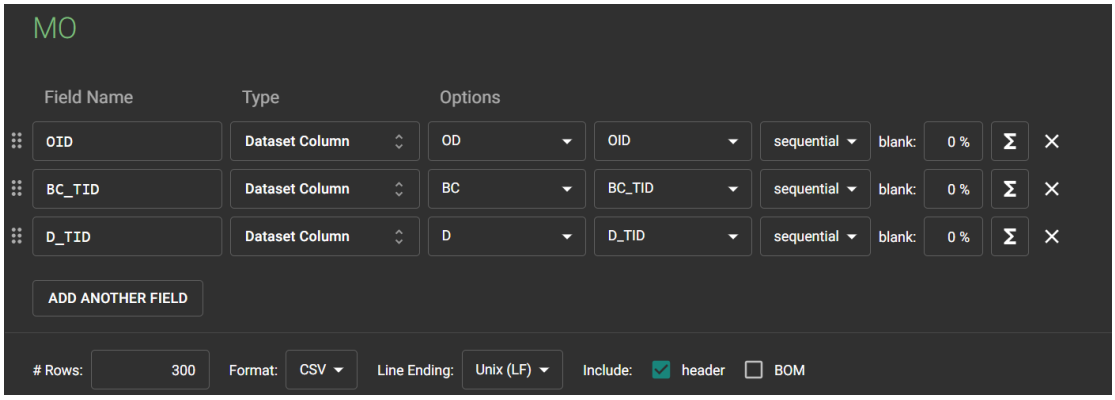
Data in Result Grid

```
75 •  select * from E;
76
```

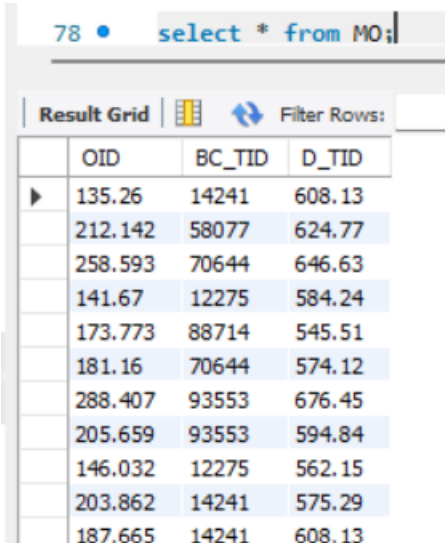
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Ce

PID	name	address	PN	BC_TID
013fcbea-e24b-44c7-bca6-bb9cb9423755	Hutcheon	490 Sommers Street	516-739-3512	29679
01465044-9fff-459e-95ad-8d2edf312c55	Higounet	82373 Sullivan Pass	653-955-5902	14241
062df801-4a03-42ba-94b6-d298b6e8a24e	Bowcock	8 Westport Lane	360-245-8265	58077
08305178-314b-416e-8694-7699d779e148	Fritschel	4 Corscot Court	731-502-0283	12275
08d2e2bd-e90b-414c-b0cc-c2312427f99c	Hanster	153 Sutteridge Point	530-870-8314	58077
0a835963-eed4-43d3-a312-2fd263e19a7d	Gallandre	94933 Victoria Street	364-810-0725	88714
0b116303-6202-4710-aa11-c684391a06b0	Gahagan	51305 2nd Parkway	845-789-1610	70644

Make Order

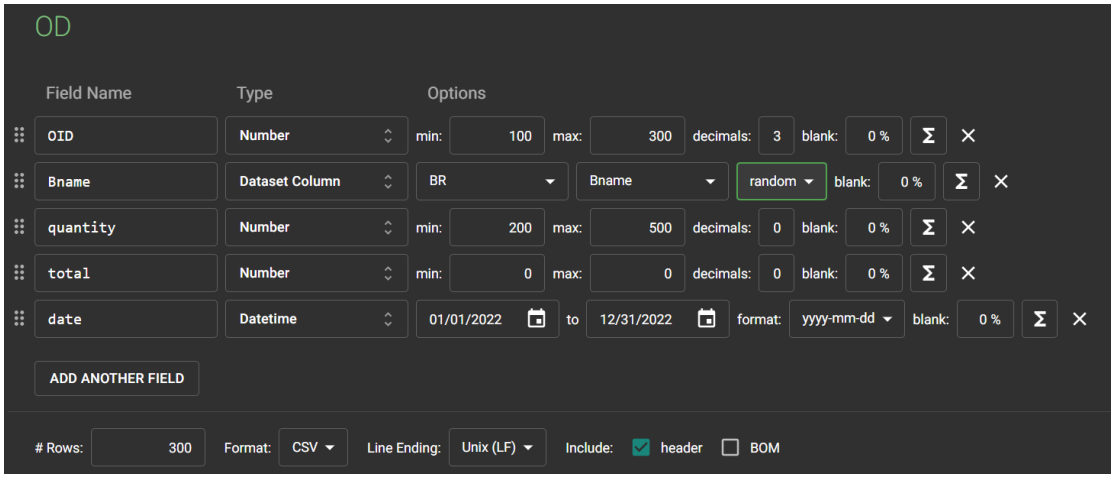


Data in Result Grid



OID	BC_TID	D_TID
135.26	14241	608.13
212.142	58077	624.77
258.593	70644	646.63
141.67	12275	584.24
173.773	88714	545.51
181.16	70644	574.12
288.407	93553	676.45
205.659	93553	594.84
146.032	12275	562.15
203.862	14241	575.29
187.665	14241	608.13

Order



Data in Result Grid (total is a derived attribute)

```
72 •    select * from OD;
```

Result Grid | Filter Rows: | Edit:

OID	Bname	quantity	total	date
100.877	Hyena, spotted	310	0	2022-03-06
101.67	Bustard, stanley	312	0	2022-05-27
102.248	Kingfisher, malachite	251	0	2022-11-01
102.321	Deer, spotted	460	0	2022-12-24
103.261	Peacock, blue	270	0	2022-12-22
105.41	Australian brush turkey	259	0	2022-01-02
105.565	Porcupine, tree	471	0	2022-12-15

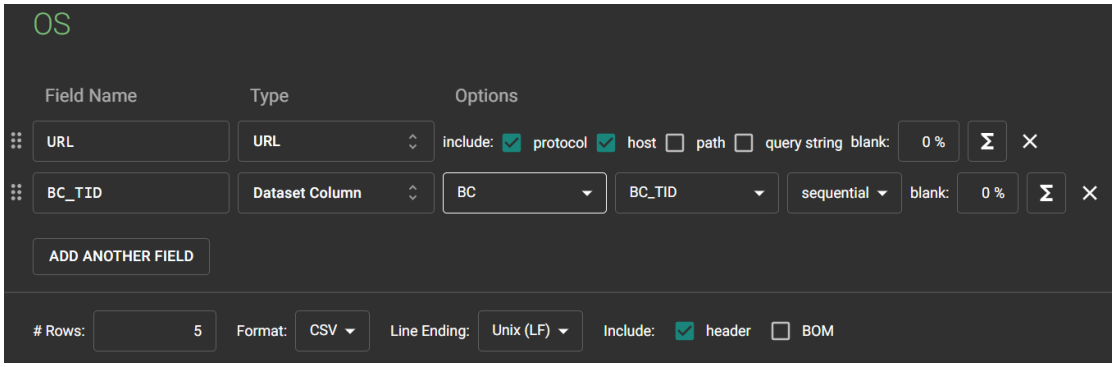
The syntax and compute result of derived attribute in MySQL

```
73 •    Update OD
74      OD JOIN BR ON OD.Bname=BR.Bname
75      SET OD.total=OD.quantity*BR.price;
76 •    select * from OD;
```

Result Grid | Filter Rows: | Edit:

OID	Bname	quantity	total	date
100.877	Hyena, spotted	310	5580	2022-03-06
101.67	Bustard, stanley	312	2808	2022-05-27
102.248	Kingfisher, malachite	251	4769	2022-11-01
102.321	Deer, spotted	460	8280	2022-12-24
103.261	Peacock, blue	270	2970	2022-12-22

Online Store



Data in Result Grid

```
77 •    select * from OS;
78
```

Result Grid | Filter Rows:

	URL	BC_TID
▶	https://go.com	29679
	http://ocn.ne.jp	46970
	https://sogou.com	47003
	https://dot.gov	70644
	https://123-reg.co.uk	88714

Part 3: Write SQL Statements to answer the following queries - 40%

1. Show the total number of transactions your database is storing and, depending on your database, the most sold/listed item or customer with the highest number of purchases.

(4 marks)

Total transaction (from OD table, the total of orders):

```
83 •    Select count(*) From OD;
```

Result Grid | Filter Rows:

	count(*)
▶	300

Total successful transaction (From MO table, the total of successful order):

```
82 •  SELECT count(*) FROM MO;
```

Result Grid | Filter Rows:

count(*)
300

Most sold Beer is **Peacock, blue** and **South African hedgehog**:

```
84 •  Select Bname, count(Bname)
85    from OD group by Bname having count(Bname);
```

Result Grid | Filter Rows: | Export:

Bname	count(Bname)
Peacock, blue	16
South African hedgehog	16
Australian brush turkey	15
Boar, wild	14
Duck, blue	12
Eastern dwarf mongoose	12
Jackal, golden	12
Porcupine, tree	12
Vulture, griffon	12
Barbet, levaillant's	11
Chickadee, black-capped	11
Dusky gull	11
Glider, sugar	11
Kingfisher, malachite	11
Seal, common	11
Covote	10

2. Write a query statement that includes "Order by" and "Group by".

(6 marks)

```
87 •  Select BC_TID, count(OID) from MO
88    group  by BC TID ORDER BY BC TID ;
```

Result Grid | Filter Rows: | Export:

BC_TID	count(OID)
12275	31
14241	31
29679	32
46970	28
47003	30
58077	22
70644	31
88714	40
93553	30
95175	25

3. Write a query statement that uses pattern matching (example: customer living in a given street, number of Johns, people with today's birthday…).

(6 marks)

```
89 •  Select * from BR where Bname like '%l%';
```

Result Grid | Filter Rows: | Edit:

Bname	price	BC_TID
Australian brush turkey	15	47003
Bandicoot, long-nosed	11	95175
Barbet, levaillant's	18	58077
Boar, wild	18	46970
Bustard, stanley	9	47003
Chickadee, black-capped	12	29679
Crane, sandhill	18	95175
Duck, blue	16	93553
Dusky gull	15	14241
Glider, sugar	11	14241

4. Show information from three tables based on criteria of your choice (hint: join).

(6 marks)

```
90 •  Select * from BC INNER JOIN BR ON BC.BC TID=BR.BC TID INNER JOIN MO ON BR.BC TID=MO.BC TID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

BC_TID	BC_name	BC_address	Bname	price	BC_TID	OID	BC_TID	D_TID
12275	Trilia	4 Vidon Point	Hyena, spotted	18	12275	208.119	12275	624.77
12275	Trilia	4 Vidon Point	Hyena, spotted	18	12275	237.765	12275	646.63
12275	Trilia	4 Vidon Point	Hyena, spotted	18	12275	177.493	12275	584.24
12275	Trilia	4 Vidon Point	Hyena, spotted	18	12275	186.775	12275	676.45
12275	Trilia	4 Vidon Point	Hyena, spotted	18	12275	280.266	12275	575.29
12275	Trilia	4 Vidon Point	Porcupine, african	19	12275	141.67	12275	584.24
12275	Trilia	4 Vidon Point	Porcupine, african	19	12275	146.032	12275	562.15
12275	Trilia	4 Vidon Point	Porcupine, african	19	12275	219.102	12275	545.51
12275	Trilia	4 Vidon Point	Porcupine, african	19	12275	279.47	12275	584.24
12275	Trilia	4 Vidon Point	Porcupine, african	19	12275	132.402	12275	545.51
12275	Trilia	4 Vidon Point	Porcupine, african	19	12275	100.877	12275	562.15

5. Create a view that includes information from the most frequent seven transactions (customer names or most sold items …).

(6 marks)

Top sold beers ranking:

1st: Peacock, blue and South African hedgehog

2nd: Australian brush turkey

3rd: Boar, wild

4th: Duck, blue and Eastern dwarf mongoose and Jackal, golden and Porcupine, tree and Vulture, griffon

```
96 •  Select Bname, count(Bname) as transactionTimes
97    from OD group by Bname having transactionTimes >11 order by transactionTimes desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Bname	transactionTimes
Peacock, blue	16
South African hedgehog	16
Australian brush turkey	15
Boar, wild	14
Duck, blue	12
Eastern dwarf mongoose	12
Jackal, golden	12
Porcupine, tree	12
Vulture, griffon	12

6. Create a set of queries that summarises the annual transactions. For example, if your transaction table is about selling product, you can create queries that:

- Shows the total number of transactions with corresponding details every month,

```
100 •  select *, month(date) as transaction_Month,count(OD.OID)AS Monthlyoder from OD
101    INNER JOIN MO ON OD.OID=MO.OID group by transaction_Month,Bname order by transaction_Month desc;
```

OID	Bname	quantity	total	date	OID	BC_TID	D_TID	transaction_Month	Monthlyoder
229.963	Barbet, levaillant's	437	7866	2022-12-04	229.963	58077	594.84	12	4
126.304	Boar, wild	437	7866	2022-12-04	126.304	46970	584.24	12	4
138.832	Bustard, stanley	321	2889	2022-12-02	138.832	47003	646.63	12	1
275.344	Coyote	431	3448	2022-12-23	275.344	70644	594.84	12	1
102.321	Deer, spotted	460	8280	2022-12-24	102.321	95175	624.77	12	2
108.833	Duck, blue	423	6768	2022-12-08	108.833	93553	575.29	12	1
208.053	Glider, sugar	407	4477	2022-12-28	208.053	14241	584.24	12	1
122.738	Hippopotamus	483	9660	2022-12-13	122.738	93553	624.77	12	1
190.432	Hyena, spotted	372	6696	2022-12-21	190.432	12275	562.15	12	1
173.773	Jackal, golden	224	3808	2022-12-26	173.773	88714	545.51	12	1
160.903	Kingfisher, malac...	429	8151	2022-12-20	160.903	93553	574.12	12	2
178.981	Lemur, sportive	288	2880	2022-12-10	178.981	46970	562.15	12	1
103.261	Peacock, blue	270	2970	2022-12-22	103.261	88714	575.29	12	3
112.434	Porcupine, african	326	6194	2022-12-09	112.434	12275	608.13	12	1
105.565	Porcupine, tree	471	6123	2022-12-15	105.565	29679	624.77	12	2
210.619	Possum, common ...	310	2170	2022-12-06	210.619	70644	574.12	12	1
174.556	Roan antelope	285	4275	2022-12-02	174.556	46970	646.63	12	2
123.675	Seal, common	426	3408	2022-12-09	123.675	58077	584.24	12	1
130.533	South African he...	394	5910	2022-12-24	130.533	70644	608.13	12	1
242.381	Tenrec, tailless	215	1935	2022-12-24	242.381	29679	676.45	12	2
203.001	Tortoise, indian star	369	5166	2022-12-12	203.001	47003	562.15	12	1
217.679	Australian brush t...	293	4395	2022-11-23	217.679	47003	676.45	11	2
293.152	Barbet, levaillant's	379	6822	2022-11-22	293.152	58077	594.84	11	1

- Shows customer purchase value per month,

```
103 •  select  month(date) as transaction_Month,sum(total)AS MonthlyPurchase from OD
104    INNER JOIN MO ON OD.OID=MO.OID group by transaction_Month order by transaction_Month desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

transaction_Month	MonthlyPurchase
12	184319
11	123254
10	100564
9	93291
8	128618
7	121159
6	150715
5	131445
4	87324
3	120348
2	129829
1	114472

- Shows name of product and number sold each month

(12 marks)

```
106 •  select Bname, quantity, month(date) as transaction_Month, sum(quantity)AS Monthlysold from OD
107    INNER JOIN MO ON OD.OID=MO.OID group by transaction_Month,Bname order by transaction_Month desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Bname	quantity	transaction_Month	Monthlysold
Barbet, levaillant's	437	12	1421
Boar, wild	437	12	1697
Bustard, stanley	321	12	321
Coyote	431	12	431
Deer, spotted	460	12	801
Duck, blue	423	12	423
Glider, sugar	407	12	407
Hippopotamus	483	12	483
Hyena, spotted	372	12	372
Jackal, golden	224	12	224
Kingfisher, malac...	429	12	816
Lemur, sportive	288	12	288
Peacock, blue	270	12	973
Porcupine, african	326	12	326
Porcupine, tree	471	12	885
Possum, common ...	310	12	310
Roan antelope	285	12	579
Seal, common	426	12	426
South African he...	394	12	394
Tenrec, tailless	215	12	434
Tortoise, indian star	369	12	369
Australian brush t...	293	11	565
Barbet, levaillant's	379	11	379