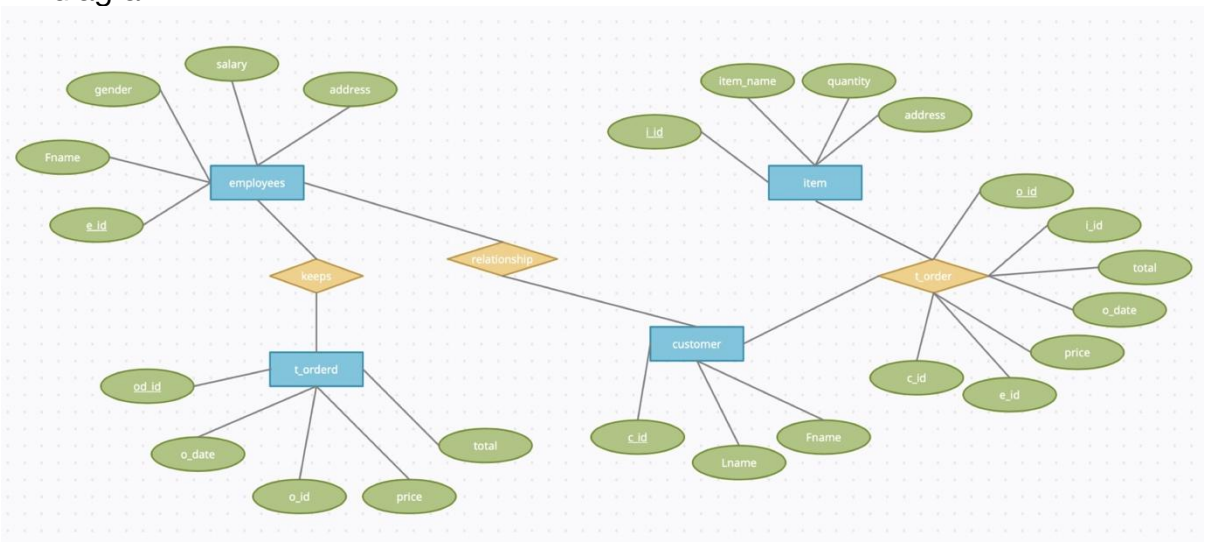
Part 1

1. Suggest a situation where you can use a database to manage and record daily transactions.

A restaurant consists of number of customers, employees and orders. Each customer offers several orders. A number of order details make up each order. Each order contains order ID, order date, the items price and the total price. The employees in the restaurant is an important part of preparing the item of the restaurant, the employees make customer orders according to order details.

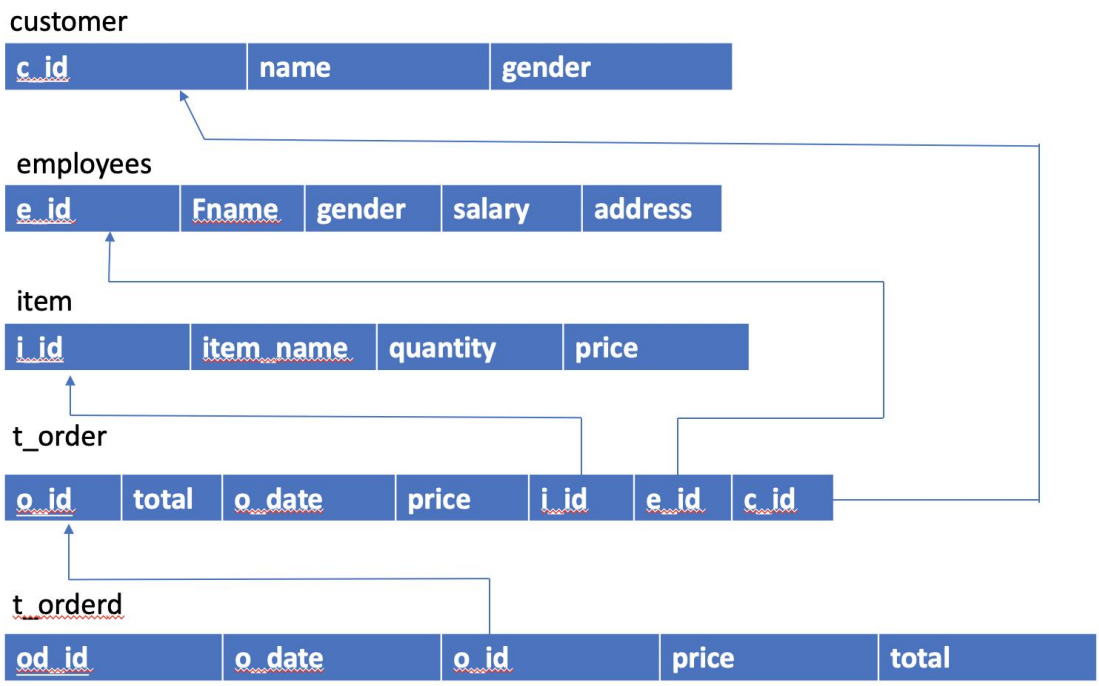
However, the employees may need to make changes to item details based on customer needs. This can help the restaurant to get a higher ratings and increase the exposure of the restaurant on the delivery app, which can lead to more orders. Each order detail is very similar to the constituent elements of order, but is different. Orders can be generated on a weekly, monthly or yearly basis for each order detail. At the same time, the order details can only show the order details of each customer, if the restaurant need to count the monthly orders or annual orders, we need the orders as the total data support. The data for each order detail can help calculate the most and least sold items for each month, as well as the difference in sales for different time periods.

2. ER diagram



3. Conceptual design into a relational model

Hansung_Restaurant



Part 2

1. Create the corresponding database using DDL and
2. Create all the necessary tables identified above using DDL

res* | Administration - Data Import/Restore

Limit to 1000 rows

```
foreign key(i_id)
references item(i_id),
foreign key(e_id)
references employees(e_id),
foreign key(c_id)
references customer(c_id)
);

create table t_orderd(
    od_id int,
    o_date datetime,
    o_id int,
    price decimal(4,2),
    total decimal(4,2),
primary key(od_id),
foreign key(o_id)
references t_order(o_id)
);
```

100% 3:57

Action Output

	Time	Action	Response
5	19:35:50	create table customer(...	0 row(s) affected
6	19:35:50	create table employees...	0 row(s) affected, 1 warning(s): 1681 Specifying number of digits for floating point data types is de
7	19:35:50	create table item(i_id i...	0 row(s) affected
8	19:35:50	create table t_order(o...	0 row(s) affected
9	19:35:50	create table t_orderd(...	0 row(s) affected

3. Populate at least three of your tables with some data using DML

```
values (51,'le','ke');
select*from customer;

insert into employees(e_id,Fname)
values(20,'shi');
select*from employees;

#insert into item(i_id,item_name)
#values (60,'Chicken');
insert into item (i_id, item_name, quantity, price) values (1, 'Chicken Mayo', 109, 10.59);
insert into item (i_id, item_name, quantity, price) values (2, 'Kimchi Dishes', 13, 11.4);
insert into item (i_id, item_name, quantity, price) values (3, 'Fried Chiecken', 562, 15.35);
```

100% 3:54 1 error found

Result Grid

c_id	Lname	Fname
51	le	ke
NULL	NULL	NULL

customer 2 | employees 3 | item 4

Action Output

	Time	Action	Response
41	14:18:23	insert into item (i_id, item_name, quantity, pr...	1 row(s) affected
42	14:18:23	insert into item (i_id, item_name, quantity, pr...	1 row(s) affected
43	14:18:23	insert into item (i_id, item_name, quantity, pr...	1 row(s) affected
44	14:18:23	select*from item LIMIT 0, 1000	3 row(s) returned

4.
1).

```
#Q2
create table customer(
    c_id int primary key,
    Lname varchar(60),
    Fname varchar(60)
);

create table employees(
    e_id int,
    Fname varchar(20),
    gender varchar(60),
    salary float(5,2),
    address varchar(60),
primary key(e_id)
);

create table item(
    i_id int,
```

100% 16:11

Action Output

	Time	Action	Response
85	19:57:27	insert into customer (c_id, Lname, Fname) values (57, 'Echlin', 'Gilbert...	1 row(s) affected
86	19:57:27	insert into customer (c_id, Lname, Fname) values (58, 'Kivlin', 'Lila')	1 row(s) affected
87	19:57:27	insert into customer (c_id, Lname, Fname) values (59, 'Wakeley', 'Hals...	1 row(s) affected
88	19:57:27	insert into customer (c_id, Lname, Fname) values (60, 'Ibbitson', 'Myrt...	1 row(s) affected

2).

```
60 •  insert into employees (e_id, Fname, gender, salary, address)
61 •  select*from employees;
```

100% 122:60

Result Grid Filter Rows: Search Export:

e_id	Fname	gender	salary	address
1	Pamela	Bigender	4327.5100000000	15 Melrose Circle
2	Rickie	Male	3894.3300000000	116 Columbus Street
3	Mellisent	Female	4967.9800000000	92 Brown Terrace
4	Asia	Female	4112.6500000000	50 Harper Junction
5	Barret	Male	5108.5100000000	68887 Sheridan Circle
6	Atlante	Female	6495.5200000000	184 Lotheville Pass
7	Obadias	Polygender	4890.1900000000	39493 Eagle Crest Place
8	Sampson	Bigender	6845.1100000000	95 Chive Road

employees 1

Action Output

	Time	Action	Response
249	21:09:53	insert into employees (e_id, Fname, gender, salary, a...	1 row(s) affected
250	21:09:53	insert into employees (e_id, Fname, gender, salary, a...	1 row(s) affected
251	21:09:53	insert into employees (e_id, Fname, gender, salary, a...	1 row(s) affected
252	21:09:53	select*from employees LIMIT 0, 1000	60 row(s) returned

3).

```
60 • insert into item (i_id, item_name, quantity, price) values (
61 • select*from item;
```

100% 18:61

Result Grid Filter Rows: Search Export:

i_id	item_name	quantity	price
3	BB Tinted	562	15.350000
4	Preparing	121	17.280000
5	DawnMist	557	17.870000
6	Pollens	293	18.320000
7	FCHICKEN DUCK	479	9.580000
8	Inspra	363	19.820000
9	Perrigo Hydroquinone	587	6.110000
10	hydrochlorothiazide	32	7.100000

item 1

Action Output

	Time	Action	Response
334	21:18:43	insert into item (i_id, item_name, quantity, price) valu...	1 row(s) affected
335	21:18:43	insert into item (i_id, item_name, quantity, price) valu...	1 row(s) affected
336	21:18:43	insert into item (i_id, item_name, quantity, price) valu...	1 row(s) affected
337	21:18:43	select*from item LIMIT 0, 1000	60 row(s) returned

4).

```
create table t_order(
    o_id int primary key,
    total decimal(60,6),
    o_date varchar(60),
    price decimal(60,6),
    i_id int,
    e_id int,
    c_id int,
foreign key(i_id)
references item(i_id),
foreign key(e_id)
references employees(e_id),
foreign key(c_id)
references customer(c_id)
);
```

100% 1:1

Action Output

	Time	Action	Response
281	22:44:57	insert into t_order (o_id, total, o_date, price, i_id, e_id, c_id) values (58,...	1 row(s) affected
282	22:44:57	insert into t_order (o_id, total, o_date, price, i_id, e_id, c_id) values (59,...	1 row(s) affected
283	22:44:57	insert into t_order (o_id, total, o_date, price, i_id, e_id, c_id) values (60,...	1 row(s) affected
284	22:44:57	select*from t_order LIMIT 0, 1000	60 row(s) returned

5).

```
create table t_orderd(
    od_id int primary key,
    o_date varchar(60),
    o_id int,
    price decimal(10,2),
    total decimal(10,2),
foreign key(o_id)
references t_order(o_id)
);
```

100% 1:1

Action Output

	Time	Action	Response
421	22:46:54	insert into t_orderd (od_id, o_date, o_id, price, total) values (57, '10/09/...	1 row(s) affected
422	22:46:54	insert into t_orderd (od_id, o_date, o_id, price, total) values (58, '09/02/...	1 row(s) affected
423	22:46:54	insert into t_orderd (od_id, o_date, o_id, price, total) values (59, '11/20/...	1 row(s) affected
424	22:46:54	insert into t_orderd (od_id, o_date, o_id, price, total) values (60, '04/04/...	1 row(s) affected

Part 3

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.

res* | customer* | employees* | item | t_order* | t_orderd

Limit to 1000 rows

```
#Part 3
#Q1.1
#Show the total number of transactions your database is storing
select sum(quantity) as 'Best Selling Dishes' from item;

#Q1.2
#Show the most sold/listed item or customer with the highest number of p
select quantity from item
where quantity = (select max(quantity) from item);

#Q2
select i_id, price from item
```

100% 1:83

Result Grid Filter Rows: Search Export:

quantity
109

item 55

Action Output

	Time	Action	Response
22	23:20:44	insert into item (i_id, item_name, quantity, price) values (1, 'Chicken Ma...	1 row(s) affected
23	23:20:53	select*from employees LIMIT 0, 1000	0 row(s) returned
24	23:23:04	select*from t_order LIMIT 0, 1000	0 row(s) returned
25	23:23:18	select quantity from item where quantity = (select max(quantity) from i...	1 row(s) returned

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

res* | customer* | employees* | item | t_order* | t_orderd

Limit to 1000 rows

```
#Q1.2
#Show the most sold/listed item or customer with the highest number of
select quantity from item
where quantity = (select max(quantity) from item);

#Q2
select i_id, price from item
group by i_id order by price DESC;

#Q3
select i_id, count(i_id) from item
where i_id = 'price';
```

100% 1:87

Result Grid Filter Rows: Search Export:

i_id	price
1	10.590000

item 56

Action Output

	Time	Action	Response
23	23:20:53	select*from employees LIMIT 0, 1000	0 row(s) returned
24	23:23:04	select*from t_order LIMIT 0, 1000	0 row(s) returned
25	23:23:18	select quantity from item where quantity = (select max(quantity) from i...	1 row(s) returned
26	23:24:10	select i_id, price from item group by i_id order by price DESC LIMIT 0, 1...	1 row(s) returned

3. Write a query statement that uses pattern matching

res* | customer* | employees* | item | t_order* | t_orderd

Limit to 1000 rows

```
where quantity = (select max(quantity) from item);

#Q2
select i_id, price from item
group by i_id order by price DESC;

#Q3
select i_id, count(i_id) from item
where i_id = 'price';

#Q4 ??????
select od_id,o_date,total from t_orderd
```

100% 1:89

Result Grid Filter Rows: Search Export:

i_id	price
60	NULL

item 28

Action Output

	Time	Action	Response
207	22:31:09	select*from t_order LIMIT 0, 1000	0 row(s) returned
208	22:34:40	select*from t_order LIMIT 0, 1000	0 row(s) returned
209	22:35:24	select quantity from item where quantity = (select m...	0 row(s) returned
210	22:35:33	select i_id, price from item group by i_id order by pri...	1 row(s) returned

4. Show information from three tables based on criteria of your choice

res | customer* | employees* | item | t_order* | t_orderd

Limit to 1000 rows

```
#Q3
select i_id, count(i_id) from item
where i_id = 'price';

#Q4
select od_id from t_orderd
inner join t_order on t_orderd.od_id=t_order.o_id
inner join item on t_orderd.o_id=item.item_name;

#Q5
select o_id, count(o_id)
as 'Most Popular Dishes' from t_order
```

100% 49:104 1 error found

Result Grid Filter Rows: Search Export:

od_id
2

5. Create a view that includes information from the most frequent seven transactions

res* | customer* | employees* | item | t_order* | t_orderd

Limit to 1000 rows

```
#Q4
select od_id from t_orderd
inner join t_order on t_orderd.od_id=t_order.o_id
inner join item on t_orderd.o_id=item.item_name;

#Q5
select o_id, count(o_id)
as 'Most Popular Dishes' from t_order
group by o_id
order by count(o_id) desc limit 7;
```

100% 1:107

Result Grid Filter Rows: Search Export:

o_id	Most Popular Dishes
1	1

Result 64

Action Output

	Time	Action	Response
59	23:44:22	insert into t_order (o_id, total, o_date, price, i_id, e_id, c_id) values (1, 8...	1 row(s) affected
60	23:47:19	insert into t_orderd (od_id, o_date, o_id, price, total) values (1, '02/16/2...	1 row(s) affected
61	23:47:30	select od_id from t_orderd inner join t_order on t_orderd.od_id=t_order....	0 row(s) returned
62	23:52:40	select o_id, count(o_id) as 'Most Popular Dishes' from t_order group by...	1 row(s) returned

6. 1). Shows the total number of transactions with corresponding details every month

```
group by o_id
order by count(o_id) desc limit 7;

#Q6.1
select month(t_orderd.o_date)
as 'month', sum(t_orderd.total),
group_concat(od_id)
as 'Monthly Transactions'from t_orderd
group by month
order by month;

#Q6.2
```

month	sum(t_orderd.total)	Monthly Transactions
6	65939.70000	65939.700000

2). Shows customer purchase value per month

```
order by month;

#Q6.2
select month(t_orderd.o_date)
as'month',sum(item.price*item.quantity)
as'Purchase Value'from item inner join t_orderd
on t_orderd.od_id=item.i_id
group by month
order by month;

#Q6.3
```

month	Purchase Value
1	20672.700000
6	37201.654000

	Time	Action	Response
104	20:12:25	insert into item (i_id, item_name, quantity, pr...	1 row(s) affected
105	20:12:25	insert into item (i_id, item_name, quantity, pr...	1 row(s) affected
106	20:12:25	insert into item (i_id, item_name, quantity, pr...	1 row(s) affected
107	20:12:25	select*from item LIMIT 0, 1000	3 row(s) returned

3). Shows name of product and number sold each month

```
order by month;


#Q6.3
select month(t_order.o_date)
as'month',
item.item_name,item.i_id,item.quantity from item
inner join t_order
on item.i_id = t_order.o_id
order by month;
```

Result Grid

month	item_name	i_id	quantity
3	Chicken Mayo	1	60