JSC "Kazakh British Technical University"
School of Mathematic and Cybernetics

Analysis of Data Bases

Laboratory Work #5

Variant #2

Prepared by: Maratuly Temirbolat

Almaty 2021

Exercise 1

Normalize to the third form the table below and build the resulted tables in SQL program

MEMBER NUM	MEMBER NAME	MEMBER ADDRESS	DINNER NUM	DINNER DATE	VENUE CODE	VENUE DESCRIPTION	FOOD CODE	FOOD DESCRIPTION
214	Peter Wong	325 Meadow Park	D0001	15-Mar-10	B01	Grand Ball Room	EN3 DE8	Stuffed crab Chocolate mousse
235	Mary Lee	123 Rose Court	D0002	15-Mar-10	B02	Petit Ball Room	EN5 DE8	Marinated steak Chocolate mousse
250	Peter Wong	9 Nine Ave	D0003	20-Mar-10	C01	Café	SO1 EN5 DE2	Pumpkin soup Marinated steak Apple pie
235	Mary Lee	123 Rose Court	D0003	20-Mar-10	C01	Café	SO1 EN5 DE2	Pumpkin soup Marinated steak Apple pie
300	Paul Lee	123 Rose Court	D0004	20-Mar-10	E10	Petit Ball Room	SA2	Apple pie

Solution:

The **First form** of the normalization is below. The procedure is just to repeat the information where the number of data in a cell appears more than one time. The columns in BOLD make a compound key.

Member Number	Member Name	Member address	**Dinnner number**	Dinner date	Venue Code	Venue Description	Food Code	Food Description
214	Peter Wong	325 Meadow Park	D0001	15-Mar-10	B01	Grand Ball Room	EN3	Stuffered crab
214	Peter Wong	325 Meadow Park	D0001	15-Mar-10	B01	Grand Ball Room	DE8	Chocolate mouse
235	Mary Lee	123 Rose Court	D0002	15-Mar-10	B02	Petit Ball Room	EN4	Marinated steak
235	Mary Lee	123 Rose Court	D0002	15-Mar-10	B02	Petit Ball Room	DE8	Chocolate mouse
250	Peter Wong	9 Nine Ave	D0003	20-Mar-10	C01	Café	S01	Pumpkin soup
250	Peter Wong	9 Nine Ave	D0003	20-Mar-10	C01	Café	EN5	Marinated steak
250	Peter Wong	9 Nine Ave	D0003	20-Mar-10	C01	Café	DE2	Apple pie
235	Mary Lee	123 Rose Court	D0003	20-Mar-10	C01	Café	S01	Pumpkin soup
235	Mary Lee	123 Rose Court	D0003	20-Mar-10	C01	Café	EN5	Marinated steak
235	Mary Lee	123 Rose Court	D0003	20-Mar-10	C01	Café	DE2	Apple pie
300	Paul Lee	123 Rose Court	D0004	20-Mar-10	E10	Petit Ball Room	DE2	Apple pie

The **Second Form** and **Third Form** of the normalization are the following. The steps to get the second form are: 1) Take all non-key attributes and all possible keys 2) Put a sign in which intersection each attribute is dependent on this key. It is called **Partial dependency.** For the Third form we just examine each table in order to find non-direct dependency from non-key attribute to key column and just extract it into new table catch **Transitive dependency.**

Second-normal form

Non-Key Attributes	Dinner_number Food_code	Member_number Dinner_number	Member_numb.	Dinner numb.	Food_code.
Member_name			✓		
Member_address			✓		
Dinner_date				✓	
Venue_code				✓	
Venue_description				✓	
Food_description					✓

Second normal form:

Member_numb, Dinner_number } Compound key

Dinner_numb. Food_code } compound key

Member_number → primary key
Member_name.
Member_address

Dinner_number → primary key
Venue_code
Venue_description.
dinner_date

Food_code → premary key
Food_description

Third normal form:

Dinner_number, Food_code } compound key → Product table

Member_number, Dinner_number } compound key → Registration table

Member_number → prem. key
Member_name
Member_address → Member Table

Food_Code → prem. key → Food table
Food_description

Dinner_number → prem key
Venue_code → Dinner table
dinner_date

Venue_code → premary key → Venue table
Venue_description

SQL part:

```sql
create table venues(
    venue_code varchar(5) constraint pk_venues PRIMARY KEY,
    venue_description varchar(30) not NULL
);
insert into venues values('B01','Grand Ball Room');
insert into venues values('B02','Petit Ball Room');
insert into venues values('C01','Cafe');
insert into venues values('E10','Petit Ball Room');
```

	venue_code	venue_description
1	B01	Grand Ball Room
2	B02	Petit Ball Room
3	C01	Cafe
4	E10	Petit Ball Room

```sql
create table dinners(
    dinner_number varchar(7) constraint pk_dinners PRIMARY KEY,
    dinner_venue_code varchar(5) UNIQUE NOT NULL,
    dinner_date date not NULL,
    constraint fk_venue_code FOREIGN KEY(dinner_venue_code) REFERENCES venues(venue_code)
);
insert into dinners values('D0001','B01','15-03-2010');
insert into dinners values('D0002','B02','15-03-2010');
insert into dinners values('D0003','C01','20-03-2010');
insert into dinners values('D0004','E10','20-03-2010');
```

	dinner_number	dinner_venue_code	dinner_date
1	D0001	B01	2010-03-15
2	D0002	B02	2010-03-15
3	D0003	C01	2010-03-20
4	D0004	E10	2010-03-20

```sql
create table food(
    food_code varchar(5) constraint pk_food PRIMARY KEY,
    food_description varchar(25) not NULL
);
insert into food values('EN3','Stuffed crab');
insert into food values('DE8','Chocolate mousse');
insert into food values('EN5','Marinated steak');
insert into food values('SO1','Pumpkin soup');
insert into food values('DE2','Apple pie');
insert into food values('SA2','Apple pie');
```

	food_code	food_description
1	EN3	Stuffed crab
2	DE8	Chocolate mousse
3	EN5	Marinated steak
4	SO1	Pumpkin soup
5	DE2	Apple pie
6	SA2	Apple pie

```sql
create table members(
    member_number integer constraint pk_member PRIMARY KEY,
    member_name varchar(40) not null,
    member_address varchar(50) not null
);
insert into members values(214,'Peter Wong','325 Meadow Park');
insert into members values(235,'Mary Lee','123 Rose Court');
insert into members values(250,'Peter Wong','9 Nine Ave');
insert into members values(300,'Paul Lee','123 Rose Court');
```

	member_number	member_name	member_address
1	214	Peter Wong	325 Meadow Park
2	235	Mary Lee	123 Rose Court
3	250	Peter Wong	9 Nine Ave
4	300	Paul Lee	123 Rose Court

```sql
create table registration(
    registration_member_number integer not NULL,
    registration_dinner_number varchar(7) not NULL,
    constraint fk_member_number FOREIGN KEY (registration_member_number) REFERENCES members(member_number),
    constraint fk_dinner_number FOREIGN KEY (registration_dinner_number) REFERENCES dinners(dinner_number),
    constraint pk_registration PRIMARY KEY (registration_member_number,registration_dinner_number)
);
insert into registration values(214,'D0001');
insert into registration values(235,'D0002');
insert into registration values(250,'D0003');
insert into registration values(235,'D0003');
insert into registration values(300,'D0004');
```

	registration_member_number	registration_dinner_number
1	214	D0001
2	235	D0002
3	250	D0003
4	300	D0004
5	235	D0003

```sql
create table products(
    product_dinnner_number varchar(7) not NULL,
    product_food_code varchar(5) not NULL,
    constraint fk_dinner_number FOREIGN KEY (product_dinnner_number) REFERENCES dinners(dinner_number),
    constraint fk_food_code FOREIGN KEY (product_food_code) REFERENCES  food(food_code),
    constraint pk_product PRIMARY KEY (product_dinnner_number,product_food_code)
);
insert into products values('D0001','EN3');
insert into products values('D0001','DE8');
insert into products values('D0002','EN5');
insert into products values('D0002','DE8');
insert into products values('D0003','SO1');
insert into products values('D0003','EN5');
insert into products values('D0003','DE2');
insert into products values('D0004','SA2');
```

	product_dinnner_number	product_food_code
1	D0001	EN3
2	D0001	DE8
3	D0002	EN5
4	D0002	DE8
5	D0003	SO1
6	D0003	EN5
7	D0003	DE2
8	D0004	SA2

Tasks:

1) Show the number of each dinner number that were done during all the days.

```
select count(d.dinner_number) as dinner_quantity, d.dinner_number,d.dinner_date
from members m join registration r
on r.registration_member_number = m.member_number
join dinners d
on d.dinner_number = r.registration_dinner_number group by dinner_number;
/*Show the number of each dinner number that were done during all the days*/
```

	dinner_quantity	dinner_number	dinner_date
1	1	D0001	2010-03-15
2	1	D0002	2010-03-15
3	1	D0004	2010-03-20
4	2	D0003	2010-03-20

2) Make the request of the first 3 letters of the name where the length of the address is higher than 10

```
select Substring(member_name,1,3) as first_three_letters_Name from members where length(member_address)>10;
/*Request the first 3 letters of the name where the length of the address is higher than 10*/
```

	first_three_letters_name
1	Pet
2	Mar
3	Pau

3) Show member name, member address, dinner number, dinner venue code, dinner date, member number of the member using double join

```
select m.member_name,m.member_address,d.dinner_number,d.dinner_venue_code,d.dinner_date,member_number
from members m join registration r
on r.registration_member_number = m.member_number
join dinners d
on d.dinner_number = r.registration_dinner_number;
/*Show member_name, member_address,dinner_number,dinner_venue_code,dinner_date of the member using double join*/
```

	member_name	member_address	dinner_number	dinner_venue_code	dinner_date	member_number
1	Peter Wong	325 Meadow Park	D0001	B01	2010-03-15	214
2	Mary Lee	123 Rose Court	D0002	B02	2010-03-15	235
3	Peter Wong	9 Nine Ave	D0003	C01	2010-03-20	250
4	Paul Lee	123 Rose Court	D0004	E10	2010-03-20	300
5	Mary Lee	123 Rose Court	D0003	C01	2010-03-20	235

4) Write the query to display member number and member name who managed to taste the biggest number of food during the all days

```
select m.member_number,m.member_name
from members m join registration r
on m.member_number = r.registration_member_number
where registration_dinner_number =
(select product_dinnner_number from products group by product_dinnner_number having count(product_dinnner_number) =
(select max(number_of_products) from (select count(product_dinnner_number) as number_of_products
from products p join food f
on p.product_food_code = f.food_code group by product_dinnner_number)a));
/*Query to display member_number and member_name who managed to taste the biggest number of food during the all days*/
```

	member_number	member_name
1	235	Mary Lee
2	250	Peter Wong

Exercise 2

Normalize to the third form the table below and build the resulted tables in SQL program

2/

OID	O_Date	CID	C_Name	C_state	PID	P_Desc	P_Price	Qty
1006	10/24/09	2	Apex	NC	7, 5, 4	Table Desk Chair	800 325 200	1, 1, 5
1007	10/25/09	6	Acme	GA	11, 4	Dresser Chair	500, 200	4 6

Un_ normalised

OID
O_Date
CID
C_Name
C_state
{PID
P_desc
P_Price
Qty }

First normal form

OID → primary key
O_date
CID
C_name
C_state

OID } compound key
PID }
P_desc
P_price
Qty.

Non-key attributes	OID + PID	OID	PID
P_desc.			✓
P_price			✓
Qty	✓		

Second-normal Form

OID → P_key
O_date
CID
C_name
C_state

PID → pr. key
P_desc
P_price

OID + } comp key
PID }
Qty

Third-normal Form

OID
O_date. → Order
CID

CID → Customer
C_name
C_state

PID → Product.
P_desc
P_price

OID
PID → Order item.
Qty

SQL part:

```sql
create table orders(
    order_id integer constraint pk_orders PRIMARY KEY,
    order_date date not null,
    customer_id integer not null,
    constraint fk_customers FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
);
insert into orders values (1006,'24-10-2009',2);
insert into orders values (1007,'25-10-2009',6);
select * from orders;
```

	order_id	order_date	customer_id
1	1006	2009-10-24	2
2	1007	2009-10-25	6

```sql
create table customers(
    customer_id integer constraint pk_customers PRIMARY KEY,
    customer_name varchar(40) not NULL,
    customer_state varchar(10) not NULL
);
insert into customers values (2,'Apex','NC');
insert into customers values (6,'Acme','GA');
select * from customers;
```

	customer_id	customer_name	customer_state
1	2	Apex	NC
2	6	Acme	GA

```sql
create table products(
    product_id integer constraint  pk_products PRIMARY KEY,
    product_description varchar(15) unique NOT NULL,
    product_price integer NOT NULL
);
insert into products values (7,'Table',800);
insert into products values (5,'Desk',325);
insert into products values (4,'Chair',200);
insert into products values (11,'Dresser',500);
```

	product_id	product_description	product_price
1	7	Table	800
2	5	Desk	325
3	4	Chair	200
4	11	Dresser	500

```sql
create table order_items(
    order_items_id integer,
    order_items_product_id integer,
    order_items_quantity integer NOT NULL,
    constraint fk_order_item_id FOREIGN KEY (order_items_id) REFERENCES orders(order_id),
    constraint fk_product_id FOREIGN KEY (order_items_product_id) REFERENCES products(product_id),
    constraint pk_order_items PRIMARY KEY(order_items_id,order_items_product_id)
);
insert into order_items values (1006,7,1);
insert into order_items values (1006,5,1);
insert into order_items values (1006,4,5);
insert into order_items values (1007,11,4);
```

	order_items_id	order_items_product_id	order_items_quantity
1	1006	7	1
2	1006	5	1
3	1006	4	5
4	1007	11	4
5	1007	4	6

Exercises:

1) Show the minimum date of the order (the oldest one) that was done

```sql
select min(order_date) as minimum_date from orders ;
/*Min date of the order (The oldest) that were done*/
```

	minimum_date
1	2009-10-24

2) Show all the descriptions of the products in UPPER Case

```sql
select UPPER(product_description) as upper_description from products order by length(product_description);
/*Show all the descriptions of the pruducts in UPPER Case*/
```

	upper_description
1	DESK
2	TABLE
3	CHAIR
4	DRESSER

3) Show order date, customer name, customer state where order date is less than 25-10-2009 using joins

```sql
select o.order_date, c.customer_name,c.customer_state
from orders o join customers c
on o.customer_id = c.customer_id
where o.order_date < '25-10-2009';
/* Show order date, customer name, customer state where order date is less than 25-10-2009 using join*/
```

	order_date	customer_name	customer_state
1	2009-10-24	Apex	NC

4) Show all the information about orders and items where the number of them is greater than 1 in correspondent order

```
select *
from orders o join order_items ot
on o.order_id = ot.order_items_id
where ot.order_items_quantity > 1;
/*Show all the information about orders and items where the number of them is greater than 1 in correspondent order */
```

	order_id	order_date	customer_id	order_items_id	order_items_product_id	order_items_quantity
1	1006	2009-10-24	2	1006	4	5
2	1007	2009-10-25	6	1007	11	4
3	1007	2009-10-25	6	1007	4	6

5) Show all the information of the order where the customer name starts with A and ends with x

```
select * from orders where orders.customer_id = (select  customer_id from customers where customer_name like  'A%x');
/* Show the all the information of the order where the customer name starts with A and ends x*/
```

	order_id	order_date	customer_id
1	1006	2009-10-24	2