employee (*ID*, *person_name*, *street*, *city*)
works (ID, *company_name*, *salary*)

Figure 1

1. (20%) Consider the employee database with two relations in Figure 1.
   (1) Write a function **avg_salary** that takes a company name as an argument and finds the average salary of employees at that company.

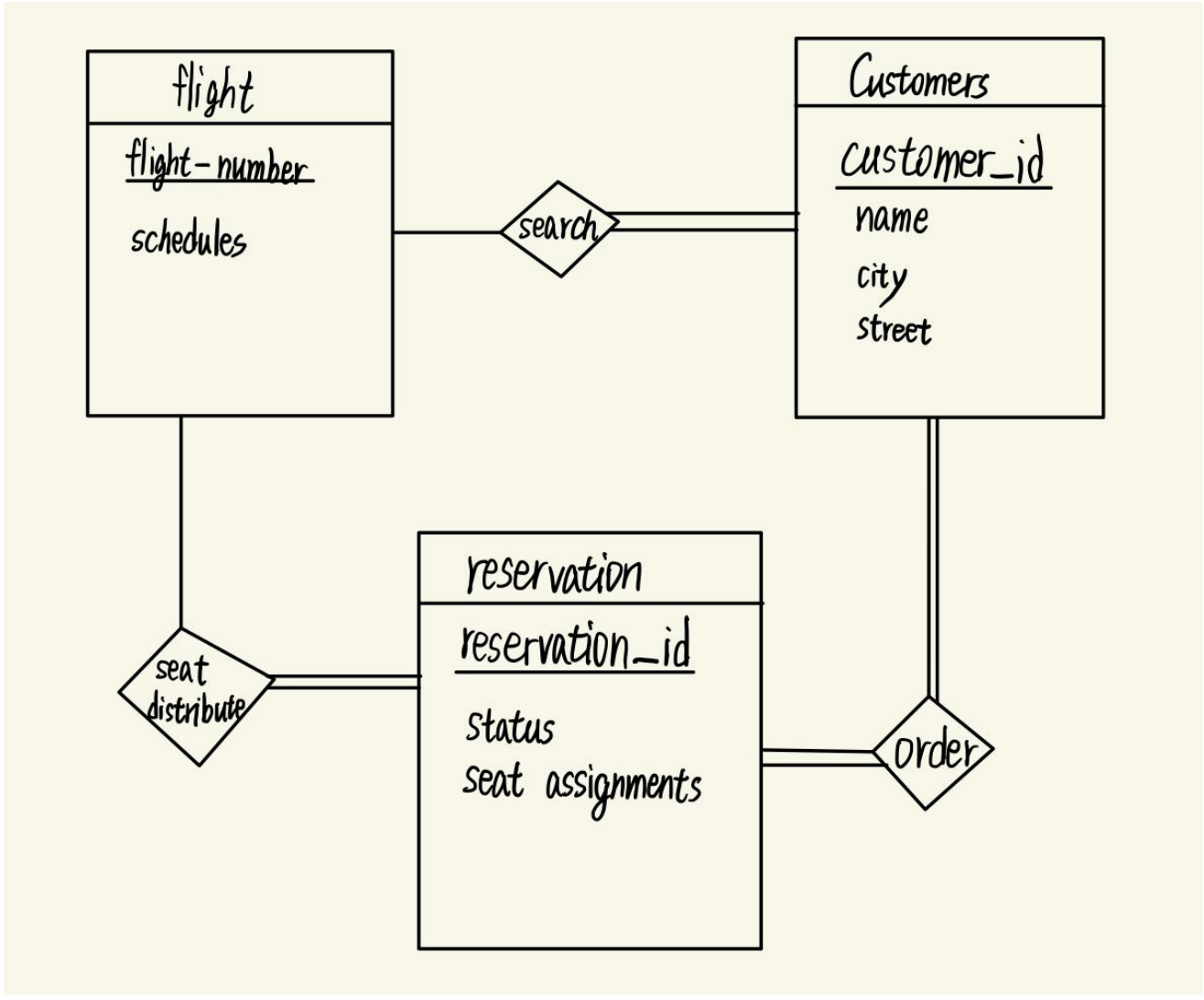
dbFunctionCreate.sql

```
1  DELIMITER $$
2
3  CREATE FUNCTION avg_salary(companyName VARCHAR(30)) RETURNS DECIMAL(10,2)
4  DETERMINISTIC
5  BEGIN
6      DECLARE ans DECIMAL(10,2);
7          SELECT AVG(salary) INTO ans FROM works
8          WHERE company_name = companyName;
9      RETURN ans;
10 END $$
11
12 DELIMITER ;  7ms
13
```

   (2) Write an SQL statement, using the **avg_salary** function, to find companies whose employees earn a higher average salary than the average salary at "FirstBank".

```
98  select distinct w.company_name as name from works as w
99  where avg_salary(w.company_name) > avg_salary('FirstBank');  2ms
```

2. (20%) Design a database using the ER-diagram for an airline. The database must represent the information of each **flight (航班)**, including its flight number and schedules (起飛降落的日期時間). The database also needs to keep track of **customers** and their **reservations** on individual flights, including the status and seat assignments. (Design the proper entity sets and relationship sets. For each entity set, represent the proper primary key and attributes.)



首先解釋entity，考慮即使單一航班也可能發生單人訂多張票的情況，reservation應該為entity，而非relation。

而customers與flight間存在可互相查詢對應的search關係，每個顧客都應該透過flight-number找到自己搭乘的flight，故search右方為total participation，flight上的座位不一定要對應到顧客 (座位未售出)，可能因時段或重大變故，出現某航空公司發出乘客數為0的航班。

對於reservation左方的seat distribute來說，所有預定應對應到flight上，不可開出沒售出的座位票，相對於flight來說，可以有座位是未售出的，因此非total participation。

對於reservation右方的order來說，所有的預定應找的到對應的顧客 (total participation)，所有顧客也應可以對應到自己的預訂紀錄，因此也為total participation。

3. (20%) Construct appropriate relational schemas for the E-R diagram in Figure 2. For each relational schema, represent the proper attributes and primary key.

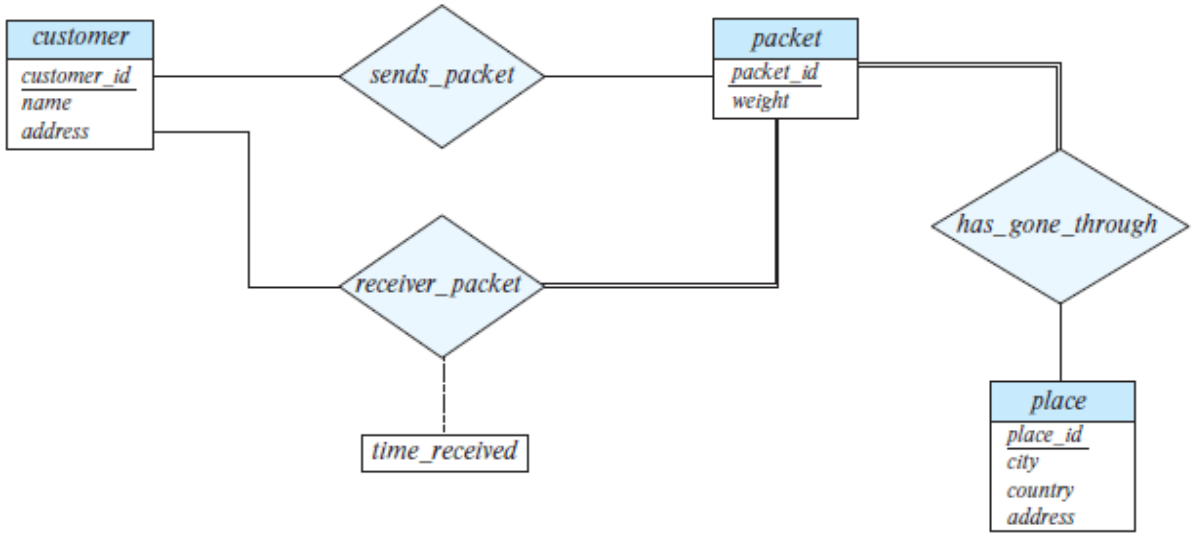


Figure 2

Sends_packet	**Sends_packet(customer_id, packet_id)**
Receiver_packet	**Receiver_packet(customer_id, packet_id, time_received)**
Has_gone_through	**Has_gone_through(packet_id, place_id)**
Customer	**Customer(customer_id, name, address)**
Packet	**Packet(packet_id, weight)**
Place	**Place(place_id, city, country, address)**

4. (20%) List two nontrivial functional dependencies satisfied by the relation in Figure 3. Explain your answer.

A	B	C
a1	b1	c1
a1	b1	c2
a2	b1	c1
a2	b1	c3

Figure 3

A -> B	**B 不被包含在 A 裡，但可以透過 A 推出唯一的 B (a1->b1, a2 ->b1)**
C -> B	**B 不包含於 A，但 C 唯一對應到 B (c1 -> b1, c2 -> b1, c3 -> b1)**

5. (20%) Consider the schema R = (A, B, C, D, E, G) and the set F of functional dependencies as follows:

{AB → CD, B → D, DE → B, DEG → AB, AC → DE}.

(1) Prove that AB is not a superkey.

1. AB 藉 AB → CD 獲得 CD
2. AB 已經有 B，不須藉 B → D 獲取 D
3. ABCD 藉 AC → DE 獲得 DE
4. ABCD 已經有 B，不須藉 DE → B 獲取 B
5. ABCDE 無法獲得 G

以上推斷，可知無法 cover 所有的 attribute，非 superkey

(2) Prove that DEG is a superkey.

1. DEG 藉 DEG → AB 獲取 AB
2. ABDEG 藉 AB → CD 獲取 C

以上推斷，可知 DEG 可以 cover 所有 attribute，為 superkey