# EE477 Database and Big Data Systems, Spring 2021
# HW2

Name: 김준범

Student ID: 20180155

Discussion Group (People with whom you discussed ideas used in your answers): None

On-line or hardcopy documents used as part of your answers: None

## Answer to Problem 1

***가독성을 위해 아래 첨자의 Table명은 앞 글자만 따서 썼습니다.**

(1) You are to find the SINs, and first and last names of employees who own an account in the branch in which they work.

$$\pi_{E.sin,E.firstName,E.LastName}( \mathbf{Employee} \bowtie_{E.firstName = C.firstName,\ E.lastName = C.lastName,\ E.branchNumber = A.branchNumber} (\mathbf{Customer} \bowtie \mathbf{Owns} \bowtie \mathbf{Accounts}) )$$

(2) The customer IDs of customers who have personal bankers in either the Vancouver or Metrotown branches (note that the personal bankers must be distinct employees as an employee only works at one branch).

$$\pi_{customerID}( \sigma_{branchName = 'Vancouver' \text{ OR } branchName = 'Metrotown'} (\mathbf{Customer} \bowtie \mathbf{PersonalBanker} \bowtie \mathbf{Employee} \bowtie \mathbf{Branch}) )$$

(3) The SINs, first and last names and salaries of employees who are both personal bankers and managers

$$\pi_{E.sin,E.firstName,E.lastName}( (\mathbf{Employee} \bowtie \mathbf{PersonalBanker}) \bowtie_{E.sin = B.managerSIN} \mathbf{Branch} )$$

(4) The SINs and salaries of employees who earn more than the manager of their branch.

$$\pi_{\text{E.sin,E.salary}}( (\text{Employee} \bowtie \text{Branch}) \bowtie_{\text{B.managerSIN = E2.sin AND E.salary > E2.salary}} (\rho_{E2(sin,fN,lN,salary,sD,bN)}\text{Employee}) )$$

(5) The customer IDs, first names, last names and incomes of customers who have an account at a branch with a budget no more than $3,200,000.

$$\pi_{\text{C.customerID,C.firstName,C.lastName,C.income}}( (\text{Customer} \bowtie \text{Owns} \bowtie \text{Account} \bowtie \sigma_{\text{budget <= 3200000}}(\text{Branch})) )$$

(6) The branch names of branches that employ at least one employee whose last name is Martin, and at least one employee whose last name is Jackson.

$$\pi_{\text{branchName}}(\sigma_{\text{lastName = 'Martin'}}(\text{Employee}) \bowtie \text{Branch}) \cap \pi_{\text{branchName}}(\sigma_{\text{lastName = 'Jackson'}}(\text{Employee}) \bowtie \text{Branch})$$

(7) The customer IDs of customers who own a joint account (an account that is owned by more than one customer)

$$\pi_{\text{O.customerID}}( \sigma_{cntCID>1}(\gamma_{accNumber,COUNT(customerID)->cntCID}(\text{Customer} \bowtie \text{Owns} \bowtie \text{Accounts})) \bowtie \text{Owns} )$$

(8) The first names, last names and birth dates of customers who own an account in the London branch, and the first names, last names and start dates of employees who work in the London branch (i.e. one query that returns one list of first and last names and dates of these 2 groups of people).

$$\rho_{\text{A(firstName,lastName,dates)}}\ \pi_{\text{C.firstName C.lastName,C.birthdata}}( (\sigma_{\text{branchName = 'London'}}(\text{Customer} \bowtie \text{Owns} \bowtie \text{Accounts} \bowtie \text{Branch}) ) \cup \rho_{\text{B(firstName,lastName,dates)}}\ \pi_{\text{E.firstName E.lastName,E.startdate}}( \sigma_{\text{branchName = 'London'}}(\text{Employee} \bowtie \text{Branch}) )$$

(9) The first and last names of customers whose first name is Steve and income is less than $40,000.

$$\pi_{\text{firstName,lastName}}(\sigma_{\text{firstName = 'steve' AND income < 40000}}(\textbf{Customer})$$

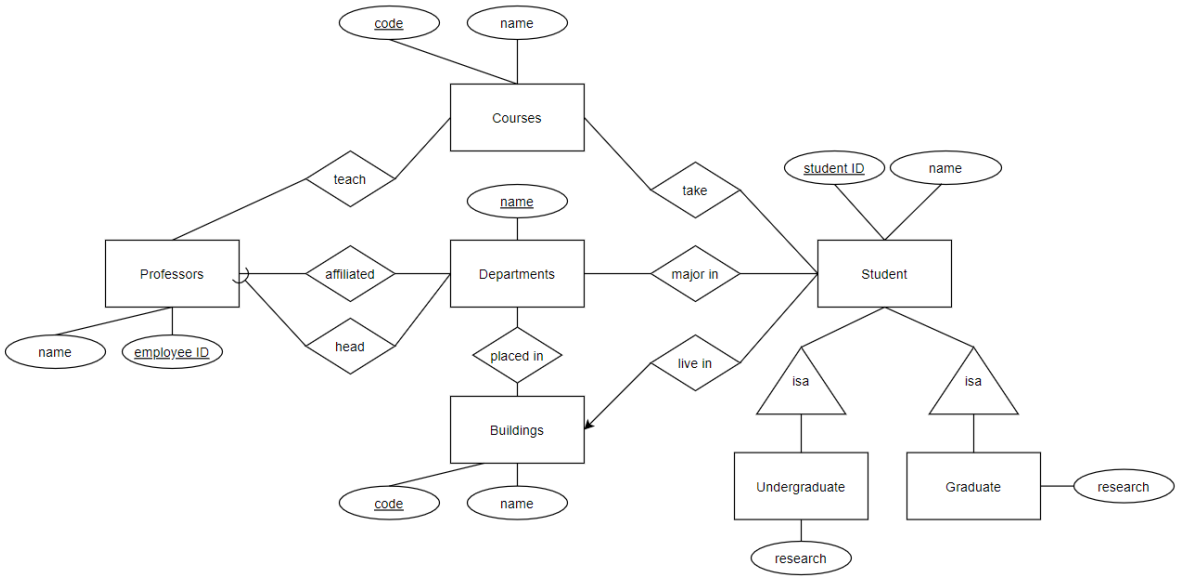
(10) The customer IDs of customers whose accounts have no transactions with amounts of which the absolute value is less than $3,000 (i.e. all their transactions are either greater than or equal to $3,000 or less than or equal to -$3,000).

$$\pi_{\text{O.customerID}}($$
$$\textbf{Owns} \bowtie$$
$$\sigma_{\text{minA<= -3000 AND maxA >=3000}} ($$
$$\gamma_{\text{accNumber,Min(amount)->minA,Max(amount)->maxA}}(\textbf{Accounts} \bowtie \textbf{Transactions}))$$
$$)$$

# Answer to Problem 2

[Task 1] Draw an E/R diagram based on the above requirements.

*Rounded arrow가 없어 Head Relation에서 거꾸로 된 round arrow를 사용했습니다.




[Task 2] Convert the E/R diagram into a schema.

Student(student_ID, name)
Professors(employee_ID, name)
Departments(name)
Buildings(code, name)
Courses(code, name)
take(student_ID, code)
major_in(student_ID, name)
live_in(student_ID, code)
placed_in(DepartmentName, BuildingsName)
head(employee_ID, name)
affiliated(employee_ID, name)
teach(employee_ID, code)
Undergraduate(student_ID, name, research)
Graduate(student_ID, name, research)

# Answer to Problem 3

[Task 1] Find all the non-trivial FDs in the table based on its schema.

**User_ID→ Gender, Age, Occupation, City_Category, Stay_In_Current_City_Years, Marital_Status**
**Product_ID → Product_Category_1, Product_Category_2, Product_Category_3**
**User_ID, Product_ID → Purchase**
**User_ID, Purchase → Product_ID**

[Task 2] For each FD in the previous task, provide an SQL query that shows that the FD indeed holds on the table.

*Table의 이름을 T라고 했습니다.

```
SELECT * FROM T
GROUP BY User_ID
HAVING COUNT (DISTINCT Gender) > 1
OR COUNT (DISTINCT Age)>1
OR COUNT (DISTINCT Occupation)>1
OR COUNT (DISTINCT City_Category)>1
OR COUNT (DISTINCT Stay_In_Current_City_Years)>1
OR COUNT (DISTINCT Marital_Status)>1 ;

SELECT * FROM T
GROUP BY Product_ID
HAVING COUNT (DISTINCT Product_Category_1) > 1
OR COUNT (DISTINCT Product_Category_2) > 1
OR COUNT (DISTINCT Product_Category_3) > 1

SELECT * FROM T
GROUP BY User_ID, Product_ID
HAVING COUNT (DISTINCT Purchase) > 1

SELECT * FROM T
GROUP BY User_ID, Purchase
HAVING COUNT (DISTINCT Product_ID) > 1
```

[Task 3] Does the table contain redundant information? Are there potential anomalies? Briefly explain with examples.

**같은 User_ID에 대해서 Gender, Age, Occupation, City_Category, Stay_In_Current_City_Years, Marital_Status 열이 모두 같은 값을 갖는다. 따라서 같은 User_ID를 가진 tuple끼리는 모두 같은 값을 가져 information들이 반복되고 있다. 또, 같은 Product_ID에 대해서 Product_Category_1, Product_Category_2, Product_Category_3 열이 같은 값을 가져 이 경우에서도 동일한 정보가 반복되어 redundant information을 가지고 있다고 볼 수 있다. 때문에 특정 User_ID를 가진 사람이 미혼에서 결혼으로 바뀐 경우, 해당 User_ID가 있는 모든 tuple의 Marital_Status를 고쳐야 하는데 이때 Update Anomaly가 발생할 수 있다.**

[Task 4] Normalize the table to remove redundant information and prevent anomalies. Write down the resulting schema and briefly explain whether the schema is in 3NF, BCNF, and/or 4NF.

Task 3에서 table이 redundant information을 가지며 이 때문에 Update Anomaly가 발생할 수 있음을 설명했다. 따라서 이 anomaly를 제거하기 위해 BCNF로 decomposition 했다(EE477 lecture7 pg57에 BCNF가 anomaly를 제거할 수 있음이 제시되어 있다).
Black Friday dataset의 key는 User_ID, Product_ID이다. 따라서 Task 1의 첫 번째 FD의 왼쪽 항은 superkey가 아니므로 violation에 해당한다. 따라서 R1 = {User_ID+}와 R2 = {User_ID and attribute not in User_ID+}로 나눌 수 있어 아래와 같이 분해된다.

R1(User_ID, Gender, Age, Occupation, City_Category, Stay_In_Current_City_Years, Marital_Status)
R2(User_ID, Product_ID, Product_Category_1, Product_Category_2, Product_Category_3, Purchase)

R1의 key는 User_ID이고 FD는 아래와 같다.
User_ID → Gender, Age, Occupation, City_Category, Stay_In_Current_City_Years, Marital_Status
FD의 왼쪽항이 User_ID이므로 더 이상 분해되지 않는다.
R2의 key는 User_ID, Product_ID이고 FD는 아래와 같다.
Product_ID → Product_Category_1, Product_Category_2, Product_Category_3
User_ID, Product_ID → Purchase
User_ID, Purchase → Product_ID

따라서 R2의 첫번째 FD의 왼쪽 항은 superkey가 아니므로 violation에 해당한다. 따라서
R3 = {ProductID+}와 R4 = {ProductID and attribute not in ProductID+}로 나눌 수 있어 아래와 같이 분해된다.

R3(Product_ID, Product_Category_1, Product_Category_2, Product_Category_3)
R4(User_ID, Product_ID, Purchase)
R3의 key는 Product_ID이고 FD는 아래와 같다.

Product_ID → Product_Category_1, Product_Category_2, Product_Category_3
FD의 왼쪽항이 Product_ID이므로 더 이상 분해되지 않는다.

R4의 key는 User_ID, Product_ID 와 User_ID, Purchase이고 FD는 아래와 같다.
User_ID, Product_ID → Purchase
User_ID, Purchase → Product_ID
FD의 왼쪽항들이 모두 key이므로 더 이상 분해되지 않는다.
따라서 anomaly를 prevent한 Resulting schema는 아래와 같다.

R1(User_ID, Gender, Age, Occupation, City_Category, Stay_In_Current_City_Years, Marital_Status)
R3(Product_ID, Product_Category_1, Product_Category_2, Product_Category_3)
R4(User_ID, Product_ID, Purchase)