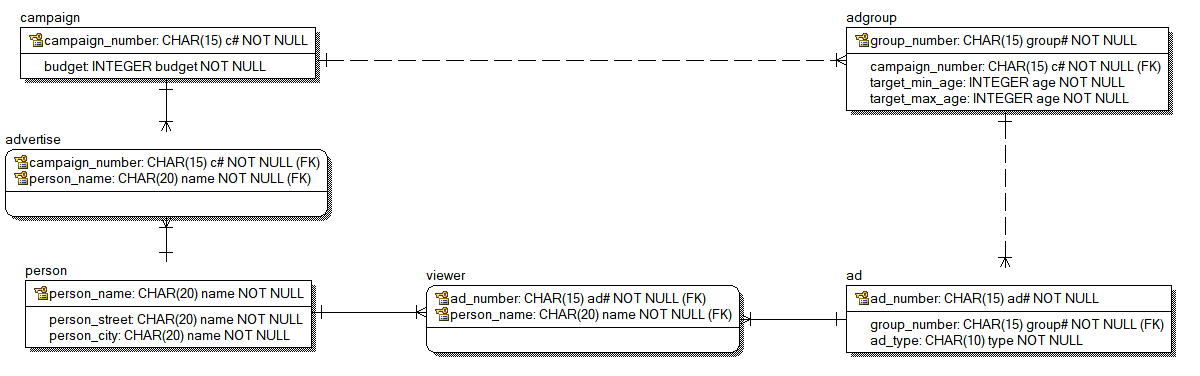
**2020 DATABASE SYSTEM PROJECT #2**

**Constructing SQL**

**학번 : 20171640**

**이름 : 박수진**

1. **Physical Mode**

****

1. Entity & Attribute

|  |  |  |
| --- | --- | --- |
| Entity | Attribute | 설명 |
| campaign | campaign\_number(PK) | 캠페인 번호 |
| budget | 캠페인에 할당된 예산 |
| adgroup | group\_number(PK) | 광고그룹 번호 |
| campaign\_number(FK) | 광고그룹이 속한 캠페인 번호 |
| target\_min\_age | 타겟의 최소 연령 |
| target\_max\_age | 타겟의 최대 연령 |
| ad | ad\_number(PK) | 광고 번호 |
| group\_number(FK) | 광고가 속한 광고그룹 번호 |
| ad\_type | 광고 타입 |
| person | person\_name(PK) | 사용자 이름 |
| person\_street | 살고 있는 거리 |
| person\_city | 살고 있는 도시 |
| advertise | campaign\_number(FK) | 집행한 캠페인 번호 |
| person\_name(FK) | 캠페인을 집행한 사용자 |
| viewer | ad\_number(FK) | 시청한 광고 번호 |
| person\_name(FK) | 광고를 시청한 사용자 |

1. Relation

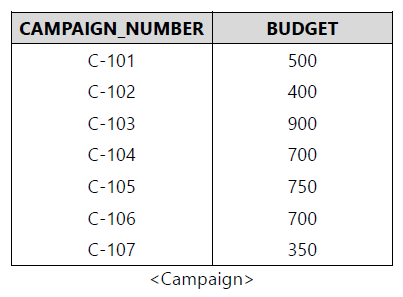
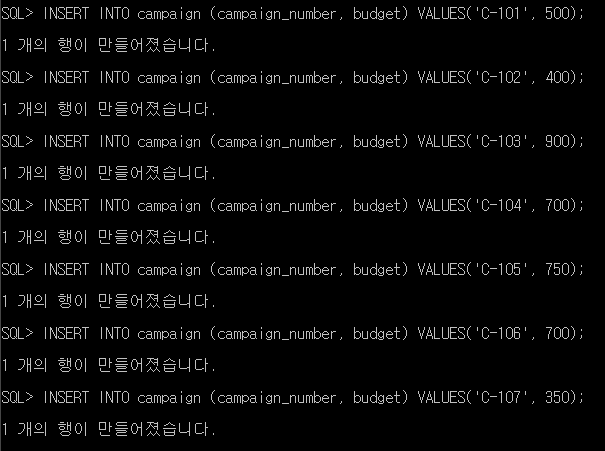
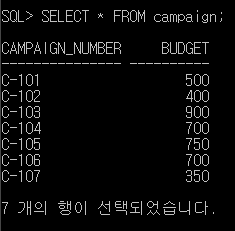
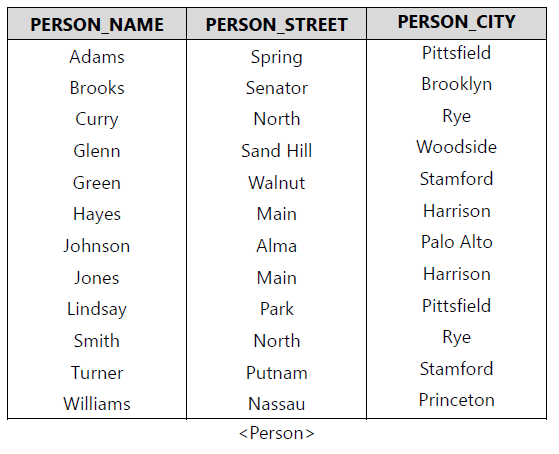
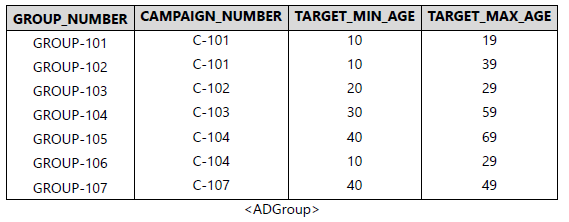
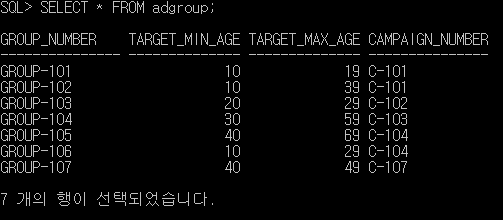
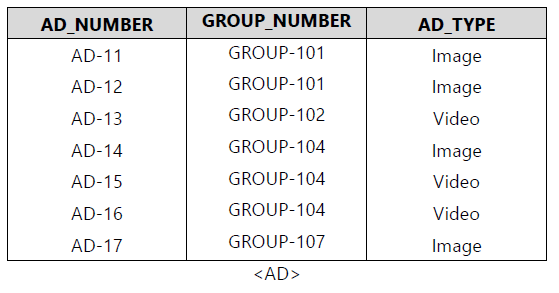
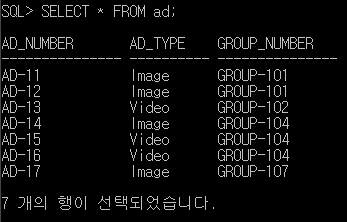
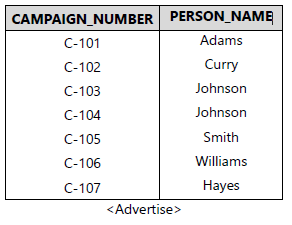
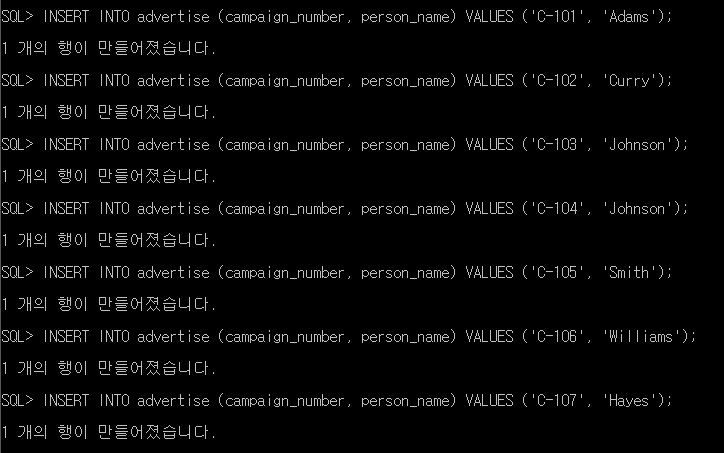
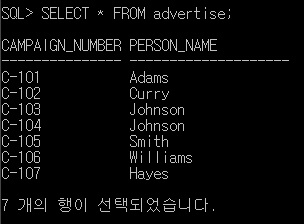
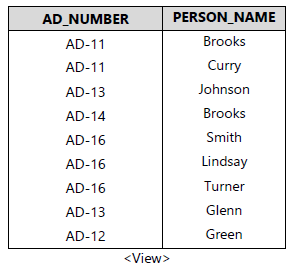
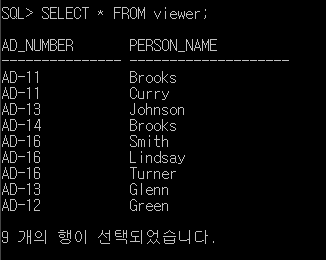
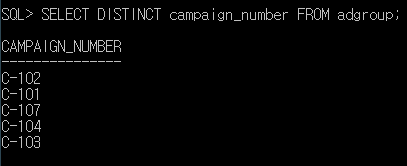
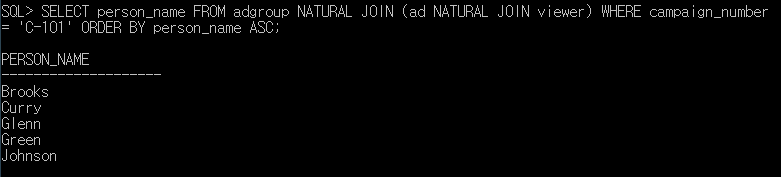
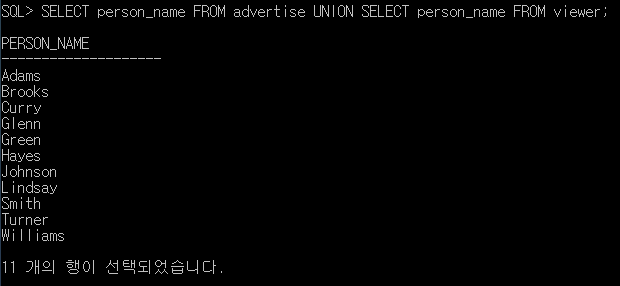
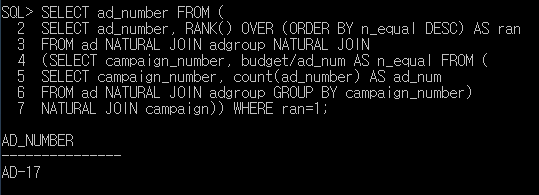
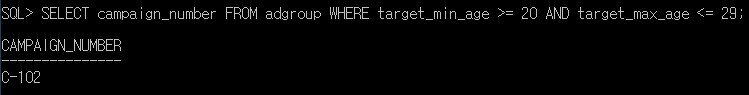
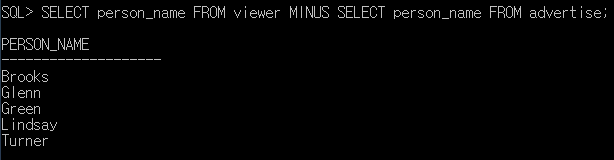
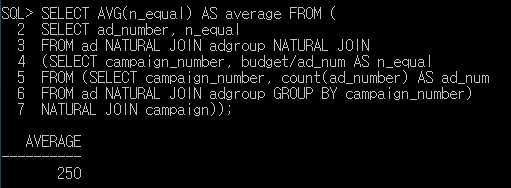
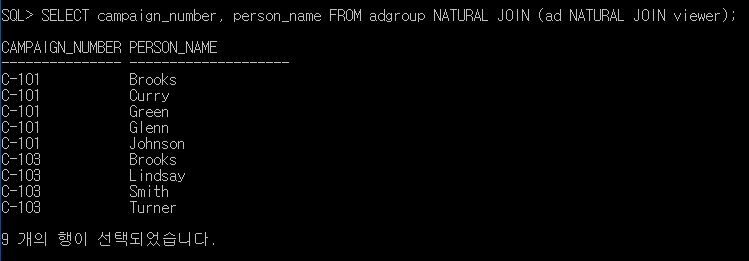
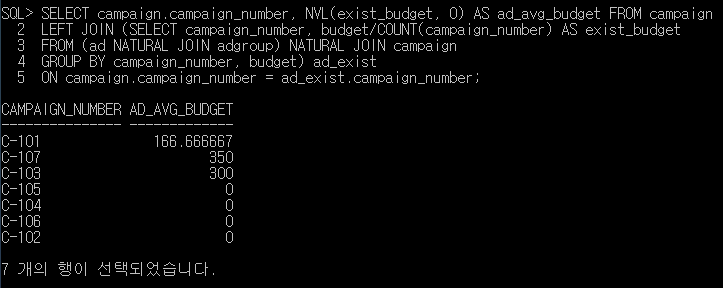
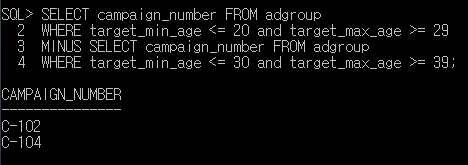
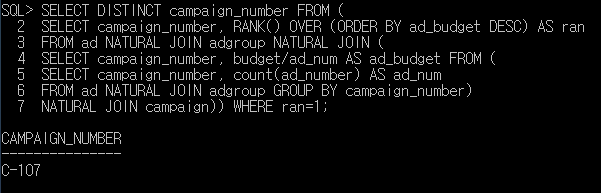
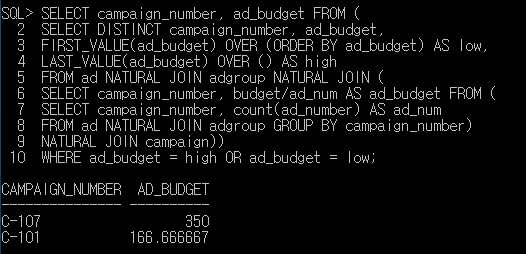
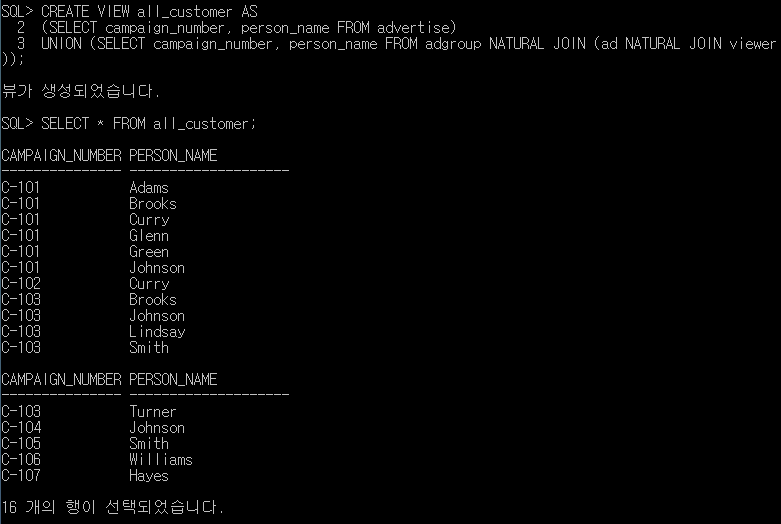
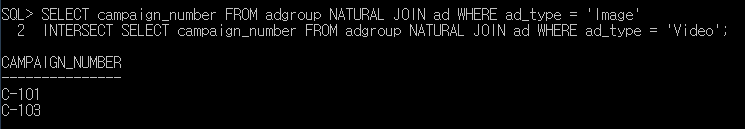
|  |  |  |  |
| --- | --- | --- | --- |
| Relation | | 설명 | |
| first\_memberof (adgroup-campaign) | | - Total participation으로 모든 adgroup들이 이 관계에 속한다.  - 하나의 캠페인 안에 여러 광고그룹이 속할 수 있으므로 adgroup과 campaign은 many to one relation이다. | |
| second\_memberof (ad-adgroup) | | - Total participation으로 모든 ad들이 이 관계에 속한다.  - 하나의 캠페인 안에 여러 광고그룹이 속할 수 있으므로 ad와 adgroup은 many to one relation이다. | |
| advertise  (person-campaign) | - 집행자가 여러 캠페인을 집행할 수 있고 캠페인은 여러 집행자를 가질 수 있으므로 many to many relation이다.  - entity로 설정하였기 때문에 추가적인 relation이 필요하다. | advertise\_campaign (advertise-campaign) | 하나의 캠페인은 여러 번 집행될 수 있으므로 many to one relation이다. |
| advertise\_person (advertise-person) | 한 명의 집행자는 여러 캠페인을 집행할 수 있으므로 many to one relation이다. |
| viewer  (person-ad) | - 시청자가 여러 광고를 시청할 수 있고 광고는 여러 시청자를 가질 수 있으므로 many to many relation이다. - entity로 설정하였기 때문에 추가적인 relation이 필요하다. | viewer\_ad (viewer-ad) | 하나의 광고는 여러 번 시청될 수 있으므로 many to one relation이다. |
| viewer\_person (viewer-person) | 한 명의 시청자는 여러 광고를 시청할 수 있으므로 many to one relation이다. |

1. Data type

|  |  |  |
| --- | --- | --- |
| Entity | Attribute | 설명 |
| campaign | campaign\_number(PK) | - 캠페인 번호는 ‘C-\*\*\*’ 형태로 들어오므로 15의 길이를 갖는 CAHR(15)로 설정하였다. - campaign\_number는 primary key이므로 NOT NULL이다. |
| budget | - 예산은 금액이기 때문에 integer로 설정하였다.  - 배정된 예산이 없어도 0으로 처리되기 때문에 NOT NULL로 설정하였다. |
| adgroup | group\_number(PK) | - 광고그룹 번호는 ‘GROUP-\*\*\*’의 형태로 들어오므로 15의 길이를 갖는 CAHR(15)로 설정하였다. - group\_number는 primary key이므로 NOT NULL이다. |
| target\_min\_age | - 특정한 나이로 입력 받으므로 INTEGER type으로 설정하였다.  - 캠페인은 타겟 연령대를 반드시 가지므로 NOT NULL로 설정하였다.  - age domain에 속한다. |
| target\_max\_age |
| ad | ad\_number(PK) | - 광고 번호는 ‘AD-\*\*’의 형태로 들어오므로 15의 길이를 갖는 CAHR(15)로 설정하였다. - ad\_number는 primary key이므로 NOT NULL이다. |
| ad\_type | - 광고 타입은 ‘Image’ 또는 ‘Video’로 입력 받으므로 10의 길이를 갖는 CHAR(10)으로 설정하였다. - 광고 타입은 반드시 사진과 비디오 둘 중 하나여야 하기 때문에 NOT NULL로 설정하였다. |
| person | person\_name(PK) | - 사용자 이름, 살고 있는 거리, 살고 있는 도시는 다양한 문자열 형태로 들어오므로 길이 20의 CHAR(20) type으로 설정하였다. - 사용자 이름은 primary key이므로 NOT NULL이다. - 사용자가 살고 있는 거리와 도시는 반드시 존재하므로 NOT NULL이다. - name domain에 속한다. |
| person\_street |
| person\_city |

1. 데이터 무결성

* 모든 entity table에는 primary key가 있다.
* 기본 키로 선택된 열은 전부 NOT NULL로 빈 값이 들어가지 않는다.
* 어떤 attribute 값들 모두 정의된 domain 범위 내에 속하고, 지정된 데이터 형식을 만족하기 때문에 데이터가 무결하다고 할 수 있다.

1. **SQL 질의문 및 결과**
2. 다음 데이터를 입력하시오 (insert)  
     
   INSERT INTO campaign (campaign\_number, budget) VALUES('C-101', 500);  
   INSERT INTO campaign (campaign\_number, budget) VALUES('C-102', 400);  
   INSERT INTO campaign (campaign\_number, budget) VALUES('C-103', 900);  
   INSERT INTO campaign (campaign\_number, budget) VALUES('C-104', 700);  
   INSERT INTO campaign (campaign\_number, budget) VALUES('C-105', 750);  
   INSERT INTO campaign (campaign\_number, budget) VALUES('C-106', 700);  
   INSERT INTO campaign (campaign\_number, budget) VALUES('C-107', 350);  
     
     
   ---------------------------------------------------------------------------------------  
     
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Adams', 'Spring', 'Pittsfield');  
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Brooks', 'Senator', 'Brooklyn');  
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Curry', 'North', 'Rye');  
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Glenn', 'Sand Hill', 'Woodside');  
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Green', 'Walnut', 'Stamford');  
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Hayes', 'Main', 'Harrison');  
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Johnson', 'Alma', 'Palo Alto');  
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Jones', 'Main', 'Harrison');  
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Lindsay', 'Park', 'Pittsfield');  
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Smith', 'North', 'Rye');  
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Turner', 'Putnam', 'Stamford');  
   INSERT INTO person (person\_name, person\_street, person\_city) VALUES('Williams', 'Nassau', 'Princeton');  
     
     
   ---------------------------------------------------------------------------------------  
     
   INSERT INTO adgroup (group\_number, campaign\_number, target\_min\_age, target\_max\_age) VALUES ('GROUP-101', 'C-101', 10, 19);  
   INSERT INTO adgroup (group\_number, campaign\_number, target\_min\_age, target\_max\_age) VALUES ('GROUP-102', 'C-101', 10, 39);  
   INSERT INTO adgroup (group\_number, campaign\_number, target\_min\_age, target\_max\_age) VALUES ('GROUP-103', 'C-102', 20, 29);  
   INSERT INTO adgroup (group\_number, campaign\_number, target\_min\_age, target\_max\_age) VALUES ('GROUP-104', 'C-103', 30, 59);  
   INSERT INTO adgroup (group\_number, campaign\_number, target\_min\_age, target\_max\_age) VALUES ('GROUP-105', 'C-104', 40, 69);  
   INSERT INTO adgroup (group\_number, campaign\_number, target\_min\_age, target\_max\_age) VALUES ('GROUP-106', 'C-104', 10, 29);  
   INSERT INTO adgroup (group\_number, campaign\_number, target\_min\_age, target\_max\_age) VALUES ('GROUP-107', 'C-107', 40, 49);  
     
     
   ---------------------------------------------------------------------------------------  
     
   INSERT INTO ad (ad\_number, group\_number, ad\_type) VALUES ('AD-11', 'GROUP-101', 'Image');  
   INSERT INTO ad (ad\_number, group\_number, ad\_type) VALUES ('AD-12', 'GROUP-101', 'Image');  
   INSERT INTO ad (ad\_number, group\_number, ad\_type) VALUES ('AD-13', 'GROUP-102', 'Video');  
   INSERT INTO ad (ad\_number, group\_number, ad\_type) VALUES ('AD-14', 'GROUP-104', 'Image');  
   INSERT INTO ad (ad\_number, group\_number, ad\_type) VALUES ('AD-15', 'GROUP-104', 'Video');  
   INSERT INTO ad (ad\_number, group\_number, ad\_type) VALUES ('AD-16', 'GROUP-104', 'Video');  
   INSERT INTO ad (ad\_number, group\_number, ad\_type) VALUES ('AD-17', 'GROUP-107', 'Image');  
     
     
   ---------------------------------------------------------------------------------------  
     
   INSERT INTO advertise (campaign\_number, person\_name) VALUES ('C-101', 'Adams');  
   INSERT INTO advertise (campaign\_number, person\_name) VALUES ('C-102', 'Curry');  
   INSERT INTO advertise (campaign\_number, person\_name) VALUES ('C-103', 'Johnson');  
   INSERT INTO advertise (campaign\_number, person\_name) VALUES ('C-104', 'Johnson');  
   INSERT INTO advertise (campaign\_number, person\_name) VALUES ('C-105', 'Smith');  
   INSERT INTO advertise (campaign\_number, person\_name) VALUES ('C-106', 'Williams');  
   INSERT INTO advertise (campaign\_number, person\_name) VALUES ('C-107', 'Hayes');  
     
     
   ---------------------------------------------------------------------------------------  
     
     
   INSERT INTO viewer (ad\_number, person\_name) VALUES ('AD-11', 'Brooks');  
   INSERT INTO viewer (ad\_number, person\_name) VALUES ('AD-11', 'Curry');  
   INSERT INTO viewer (ad\_number, person\_name) VALUES ('AD-13', 'Johnson');  
   INSERT INTO viewer (ad\_number, person\_name) VALUES ('AD-14', 'Brooks');  
   INSERT INTO viewer (ad\_number, person\_name) VALUES ('AD-16', 'Smith');  
   INSERT INTO viewer (ad\_number, person\_name) VALUES ('AD-16', 'Lindsay');  
   INSERT INTO viewer (ad\_number, person\_name) VALUES ('AD-16', 'Turner');  
   INSERT INTO viewer (ad\_number, person\_name) VALUES ('AD-13', 'Glenn');  
   INSERT INTO viewer (ad\_number, person\_name) VALUES ('AD-12', 'Green');  
     
     
   
3. 광고그룹을 가지고 있는 캠페인 번호를 중복되지 않게 나열하라.  
   SELECT DISTINCT campaign\_number FROM adgroup;  
   
4. 캠페인 C-101에 포함된 광고를 본 사람들의 이름을 알파벳 순서대로 나열하라.  
   SELECT person\_name FROM adgroup NATURAL JOIN (ad NATURAL JOIN viewer) WHERE campaign\_number = 'C-101' ORDER BY person\_name ASC;  
   
5. 캠페인을 집행 또는 광고를 시청한 사용자를 나열하라.  
   SELECT person\_name FROM advertise UNION SELECT person\_name FROM viewer;  
   
6. 캠페인을 집행 그리고 광고를 시청한 사용자를 나열하라.  
   SELECT person\_name FROM advertise INTERSECT SELECT person\_name FROM viewer;  
   
7. 캠페인의 예산이 캠페인에 포함된 광고의 개수로 n등분 된다고 가정했을 때, 가장 많은 예산이 소모된 광고를 구하라.  
   SELECT ad\_number FROM (  
   SELECT ad\_number, RANK() OVER (ORDER BY n\_equal DESC) AS ran   
   FROM ad NATURAL JOIN adgroup NATURAL JOIN   
   (SELECT campaign\_number, budget/ad\_num AS n\_equal FROM (  
   SELECT campaign\_number, count(ad\_number) AS ad\_num   
   FROM ad NATURAL JOIN adgroup GROUP BY campaign\_number)   
   NATURAL JOIN campaign)) WHERE ran=1;  
   
8. 타겟 연령층이 20대인 광고그룹을 포함하는 캠페인을 나열하라.  
   SELECT campaign\_number FROM adgroup WHERE target\_min\_age >= 20 AND target\_max\_age <= 29;  
   
9. 캠페인을 집행한 적이 없지만 광고를 시청한 사용자를 나열하라.  
   SELECT person\_name FROM viewer MINUS SELECT person\_name FROM advertise;  
   
10. 캠페인의 예산이 캠페인에 포함된 광고의 개수로 n등분 된다고 가정했을 때, 광고 하나에 소모되는 평균 예산을 구하여라.  
    SELECT AVG(n\_equal) AS average FROM (  
    SELECT ad\_number, n\_equal   
    FROM ad NATURAL JOIN adgroup NATURAL JOIN   
    (SELECT campaign\_number, budget/ad\_num AS n\_equal   
    FROM (SELECT campaign\_number, count(ad\_number) AS ad\_num   
    FROM ad NATURAL JOIN adgroup GROUP BY campaign\_number)   
    NATURAL JOIN campaign));  
    
11. 각 캠페인 별 시청 사용자를 나열하라.  
    SELECT campaign\_number, person\_name FROM adgroup NATURAL JOIN (ad NATURAL JOIN viewer);  
    
12. 캠페인의 예산이 캠페인에 포함된 광고의 개수로 n등분 된다고 가정했을 때, 각 캠페인 별로 하나의 광고에 소모되는 평균 예산을 구하여라.  
    SELECT campaign.campaign\_number, NVL(exist\_budget, 0) AS ad\_avg\_budget FROM campaign   
    LEFT JOIN (SELECT campaign\_number, budget/COUNT(campaign\_number) AS exist\_budget   
    FROM (ad NATURAL JOIN adgroup) NATURAL JOIN campaign   
    GROUP BY campaign\_number, budget) ad\_exist   
    ON campaign.campaign\_number = ad\_exist.campaign\_number;  
    
13. 타겟 연령층이 20대를 포함하고 30대는 포함하지 않는 광고그룹을 포함하는 캠페인을 나열하라.  
    SELECT campaign\_number FROM adgroup   
    WHERE target\_min\_age <= 20 and target\_max\_age >= 29   
    MINUS SELECT campaign\_number FROM adgroup   
    WHERE target\_min\_age <= 30 and target\_max\_age >= 39;  
    
14. 캠페인의 예산이 캠페인에 포함된 광고의 개수로 n등분 된다고 가정했을 때, 가장 많은 예산이 소모된 광고를 포함하는 캠페인을 구하여라.  
    SELECT DISTINCT campaign\_number FROM (  
    SELECT campaign\_number, RANK() OVER (ORDER BY ad\_budget DESC) AS ran   
    FROM ad NATURAL JOIN adgroup NATURAL JOIN (  
    SELECT campaign\_number, budget/ad\_num AS ad\_budget FROM (  
    SELECT campaign\_number, count(ad\_number) AS ad\_num   
    FROM ad NATURAL JOIN adgroup GROUP BY campaign\_number)   
    NATURAL JOIN campaign)) WHERE ran=1;  
    
15. Stamford 도시에 살고 있는 사용자가 시청한 광고를 포함하는 캠페인을 모두 나열하라.  
    SELECT campaign\_number FROM (ad NATURAL JOIN (SELECT ad\_number FROM viewer NATURAL JOIN person WHERE person\_city = 'Stamford')) NATURAL JOIN adgroup;  
    
16. 알파벳 ‘N’으로 시작하는 거리에 살고 있는 사용자가 집행한 캠페인을 모두 나열하라.  
    SELECT campaign\_number FROM (person NATURAL JOIN advertise) WHERE person\_street LIKE 'N%';  
    
17. 캠페인의 예산이 캠페인에 포함된 광고의 개수로 n등분 된다고 가정했을 때, 가장 많은 예산이 소모된 광고를 포함하는 캠페인과, 가장 적은 예산이 소모된 광고를 포함하는 캠페인을 나열하라. (예산이 0인 광고는 고려하지 않는다.)  
    SELECT campaign\_number, ad\_budget FROM (  
    SELECT DISTINCT campaign\_number, ad\_budget,   
    FIRST\_VALUE(ad\_budget) OVER (ORDER BY ad\_budget) AS low,   
    LAST\_VALUE(ad\_budget) OVER () AS high   
    FROM ad NATURAL JOIN adgroup NATURAL JOIN (  
    SELECT campaign\_number, budget/ad\_num AS ad\_budget FROM (  
    SELECT campaign\_number, count(ad\_number) AS ad\_num   
    FROM ad NATURAL JOIN adgroup GROUP BY campaign\_number)   
    NATURAL JOIN campaign))   
    WHERE ad\_budget = high OR ad\_budget = low;  
    
18. 캠페인 번호와 캠페인을 집행했거나, 해당 캠페인에 포함된 광고를 시청한 사용자의 이름으로 구성된 View를 작성하라. (단 View의 이름은 all\_customer이다.)  
    CREATE VIEW all\_customer AS   
    (SELECT campaign\_number, person\_name FROM advertise)   
    UNION (SELECT campaign\_number, person\_name FROM adgroup NATURAL JOIN (ad NATURAL JOIN viewer));  
    
19. 17에서 생성된 View를 이용하여 Harrison 도시에 살고 있는 모든 고객을 나열하라.  
    SELECT person\_name FROM all\_customer NATURAL JOIN person WHERE person\_city = 'Harrison';  
    
20. 이미지와 비디오 광고를 모두 포함하고 있는 캠페인을 나열하라.  
    SELECT campaign\_number FROM adgroup NATURAL JOIN ad WHERE ad\_type = 'Image'   
    INTERSECT SELECT campaign\_number FROM adgroup NATURAL JOIN ad WHERE ad\_type = 'Video';  
    
21. 이미지 또는 비디오 광고로만 구성된 캠페인을 나열하라. (이미지와 비디오 광고가 같이 있으면 안됨.)  
    ((SELECT campaign\_number FROM adgroup NATURAL JOIN ad WHERE ad\_type = 'Image')   
    MINUS (SELECT campaign\_number FROM adgroup NATURAL JOIN ad WHERE ad\_type = 'Video'))   
    UNION ((SELECT campaign\_number FROM adgroup NATURAL JOIN ad WHERE ad\_type = 'Video')   
    MINUS (SELECT campaign\_number FROM adgroup NATURAL JOIN ad WHERE ad\_type = 'Image'));  
    