

〈7주차 실습〉

# 서브쿼리, 테이블 변경

---

Database Programming



실습 테이블



Subquery




테이블 변경




실습 과제



EMPLOYEES

 ID: INT
Name: CHAR(50)
Department: CHAR(20)
Budget: INT


BOATS

 Bid: INT
Bname: VARCHAR2(100)
Color: VARCHAR2(50)


RESERVES

 Bid: INT (FK)
 Sid: INT (FK)
Day: DATE


SAILORS

 Sid: INT
Sname: CHAR(50)
Rating: INT
Age: NUMBER



WORKS

 WorkID: INT
Title: VARCHAR2(55)
Medium: VARCHAR2(100)
Description: VARCHAR2(1000)
Copy: VARCHAR2(100)
ArtistID: INT (FK)


ARTISTS

 ArtistID: INT
LastName: CHAR(25)
FirstName: CHAR(25)
Nationality: CHAR(30)
DateOfBirth: NUMBER(4)
DateDeceased: NUMBER(4)

CUSTOMERS\_ARTISTS\_INT

 ArtistID: INT (FK)
 CustomerID: INT (FK)

CUSTOMERS

 CustomerID: INT
LastName: CHAR(25)
FirstName: CHAR(25)

## EMPLOYEES 테이블에 대한 데이터베이스 열 특성

EMPLOYEES		
Column Name	Type	Key & Null Status
Id	INT	Primary Key
Name	CHAR(50)	Not null
Department	CHAR(20)	Not null
Budget	INT	

## EMPLOYEES 테이블 데이터

EMPLOYEES			
Id	Name	Department	Budget
1	King	SALES	100
2	Clark	OPERATIONS	80
3	Ford	ACCOUNTING	20
4	Smith	RESEARCH	70
5	Adams	MARKETING	40

## ■ BOATS 테이블에 대한 데이터베이스 열 특성

BOATS		
Column Name	Type	Key & Null Status
Bid	INT	Primary Key
Bname	VARCHAR2(100)	Not null
Color	VARCHAR2(50)	Not null

## ▣ BOATS 테이블 데이터

BOATS		
Bid	Bname	Color
101	Interlake	Blue
102	Interlake	Red
103	Clipper	Green
104	Marine	Blue

## ■ SAILORS 테이블에 대한 데이터베이스 열 특성

SAILORS		
Column Name	Type	Key & Null Status
Sid	INT	Primary Key
Sname	CHAR(50)	Not null
Rating	INT	Not null
Age	NUMBER	



## ■ SAIORS 테이블 데이터

SAIORS			
Sid	Sname	Rating	Age
22	Dustin	7	45.2
29	Brutus	1	33.0
31	Lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35.0
64	Horatio	7	35.0
71	Zorba	10	16.0
74	Tom	7	35.0
85	Art	3	25.5
95	Bob	3	63.5

## RESERVES 테이블에 대한 데이터베이스 열 특성

RESERVES		
Column Name	Type	Key & Null Status
Sid	INT	Primary Key, Foreign Key <Sailor(Sid)>
Bid	INT	Primary Key, Foreign Key <Boats(Bid)>
Day	DATE	Not null

## RESERVES 테이블 데이터

RESERVES		
Sid	Bid	Day
22	101	10/10/98
22	102	10/10/98
22	103	10/8/98
22	104	10/7/98
31	102	11/10/98
31	103	11/6/98
31	104	11/12/98
64	101	9/5/98
64	102	9/8/98
74	103	9/8/98

## ■ ARTISTS 테이블에 대한 데이터베이스 열 특성

ARTISTS		
Column Name	Type	Key & Null Status
ArtistID	INT	Primary Key
LastName	CHAR(25)	Not null
FirstName	CHAR(25)	Not null
Nationality	CHAR(30)	
DateOfBirth	NUMBER(4)	
DateDeceased	NUMBER(4)	

## ■ ARTISTS 테이블 데이터

ARTISTS					
ArtistID	LastName	FirstName	Nationality	DateOf Birth	DateDeceased
1	Miro	Joan	Spanish	1893	1983
2	Kandinsky	Wassily	Russian	1866	1944
3	Klee	Paul	German	1879	1940
4	Matisse	Henri	French	1869	1954
5	Chagall	Marc	French	1887	1985
11	Sargent	John Singer	United States	1856	1925
17	Tobey	Mark	United States	1890	1976
18	Horiuchi	Paul	United States	1906	1999
19	Graves	Morris	United States	1920	2001

## ■ WORKS 테이블에 대한 데이터베이스 열 특성

WORKS			
Column Name	Type	Key & Null Status	Remarks
WorkID	INT	Primary Key	
Title	VARCHAR2(55)	Not null	
Medium	VARCHAR2(100)		
Description	VARCHAR2(1000)		DEFAULT value= 'Unknown provenance'
Copy	VARCHAR2(100)	Not null	
ArtistID	INT	Foreign Key <Artists(ArtistID)>	

## ■ WORKS 테이블 데이터

WORKS					
WorkID	Title	Medium	Description	Copy	ArtistID
500	Memories IV	Casein rice paper collage	31 x 24.8 in	Unique	18
501	Surf and Bird	High Quality Limited Print	Northwest School Expressionist style	142/500	19
502	The tilled Field	High Quality Limited Print	Early Surrealist style	788/1000	1
503	La Lecon de Ski	High Quality Limited Print	Surrealist style	353/500	1
504	On White II	High Quality Limited Print	Bauhaus style of Kandinsky	435/500	2
505	Woman with a Hat	High Quality Limited Print	A very colorful Impressionist piece	596/750	4
506	The Woven World	Color lithograph	Signed	17/750	17
507	Night Bird	Watercolor on Paper	50 x 72.5 cm-signed	Unique	19
508	Der Blaue Reiter	High Quality Limited Print	"The Blue Rider"-Early Pointilism influence	236/1000	2
509	Angelus Novus	High Quality Limited Print	Bauhaus style of Klee	659/750	3
510	The Dance	High Quality Limited Print	An Impressionist masterpiece	734/1000	4

## ■ CUSTOMERS 테이블에 대한 데이터베이스 열 특성

CUSTOMERS			
Column Name	Type	Key & Null Status	Remarks
CustomerID	INT	Primary Key	
Lastname	CHAR(25)	Not null	
Firstname	CHAR(25)	Not null	

## ■ CUSTOMERS 테이블 데이터

CUSTOMERS		
CustomerID	Lastname	Firstname
1000	Janes	Jeffrey
1001	Smith	David
1015	Twilight	Tiffany



## ■ CUSTOMERS\_ARTISTS\_INT 테이블에 대한 데이터베이스 열 특성

CUSTOMERS_ARTISTS_INT			
Column Name	Type	Key & Null Status	Remarks
ArtistID	INT	Primary Key	
CustomerID	INT	Primary Key	

## ■ CUSTOMERS\_ARTISTS\_INT 테이블 데이터

CUSTOMERS_ARTISTS_INT	
ArtistID	CustomerID
1	1001
11	1000
11	1001
11	1015
17	1015

## ■ Non-Correlated Subquery

```
SQL> SELECT A.LastName  
2   FROM ARTISTS A  
3   WHERE A.ArtistID IN  
4         (SELECT W.ArtistID  
5           FROM WORKS W  
6           WHERE W.Title = 'The Dance');
```

```
LASTNAME
```

```
-----
```

```
Matisse
```

## ■ Correlated Subquery – 중복행 찾기

**INSERT INTO WORKS(WorkID, Title, Medium, Description, Copy, ArtistID)**  
**VALUES (999, 'Memories IV', 'Casein rice paper collage', '31 x 24.8 in.', 'Unique', 18);**



```
SQL> SELECT W1.WorkID, W1.Title
2 FROM WORKS W1
3 WHERE W1.Title IN
4       (SELECT W2.Title
5         FROM WORKS W2
6         WHERE W1.Title = W2.Title
7         AND W1.WorkID <> W2.WorkID);
```

WORKID	TITLE
--------	-------

999	Memories IV
-----	-------------

500	Memories IV
-----	-------------

## ■ Checking Functional Dependencies

INSERT INTO EMPLOYEES VALUES (6, 'Adams', 'MARKETING', '88');



```
SQL> SELECT E1.Department, E1.Budget
  2  FROM EMPLOYEES E1
  3  WHERE E1.Department IN
  4         (SELECT E2.Department
  5            FROM EMPLOYEES E2
  6            WHERE E1.Department = E2.Department
  7            AND E1.Budget <> E2.Budget);
```

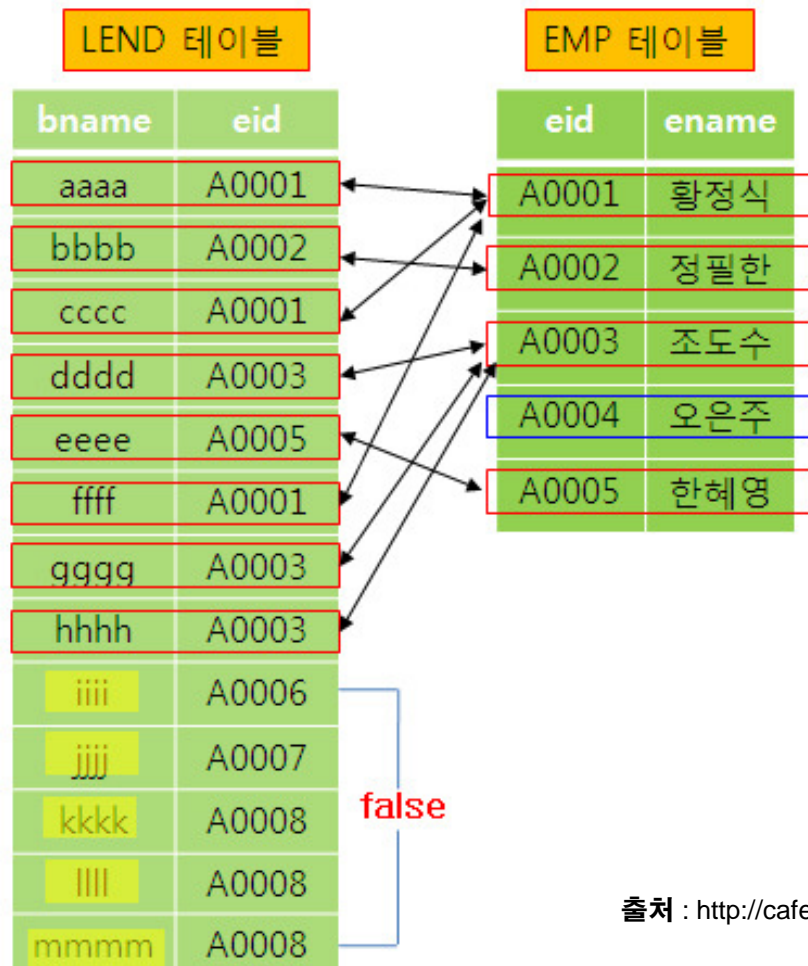
DEPARTMENT	BUDGET
MARKETING	40
MARKETING	88

## ■ Checking Functional Dependencies

```
SQL> SELECT E1.Department, E1.Budget
  2  FROM EMPLOYEES E1
  3  WHERE EXISTS
  4         (SELECT *
  5          FROM EMPLOYEES E2
  6          WHERE E1.Department = E2.Department
  7          AND E1.Budget <> E2.Budget);
```

DEPARTMENT	BUDGET
MARKETING	40
MARKETING	88

## NOT EXISTS 예시



책을 대출한 비정규직 직원들의 eid(사원 아이디)를 조회하고자 할때, 예제가 그나마 가장 적절할듯...

\* 쿼리

```
select l.bname from lend l where NOT EXISTS ( select *  
from emp e where l.eid = e.eid );
```

--> 결과는 LEND 테이블의 노란색 음영처리된 데이터이다. 양방향 화살표는 true인 eid들이다.

==== 쉽게 이해하는 나만의 방법 ====

: a.컬럼 = b.컬럼 즉, a.컬럼과 b.컬럼을 조인했을때 조인이되지 않은 (false) 컬럼의 데이터를 모두 조회한다.

- 모든 CUSTOMERS가 관심을 가지고 있는 작가를 검색

```
SELECT A.FirstName, A.LastName
FROM ARTISTS A
WHERE NOT EXISTS
    (SELECT C.CustomerID
     FROM CUSTOMERS C
     WHERE NOT EXISTS
        (SELECT CAI.CustomerID
         From CUSTOMERS_ARTISTS_INT CAI
         WHERE C.CustomerID = CAI.CustomerID
         AND A.ArtistID = CAI.ArtistID));
```

FIRSTNAME	LASTNAME
John Singer	Sargent

Artists Table

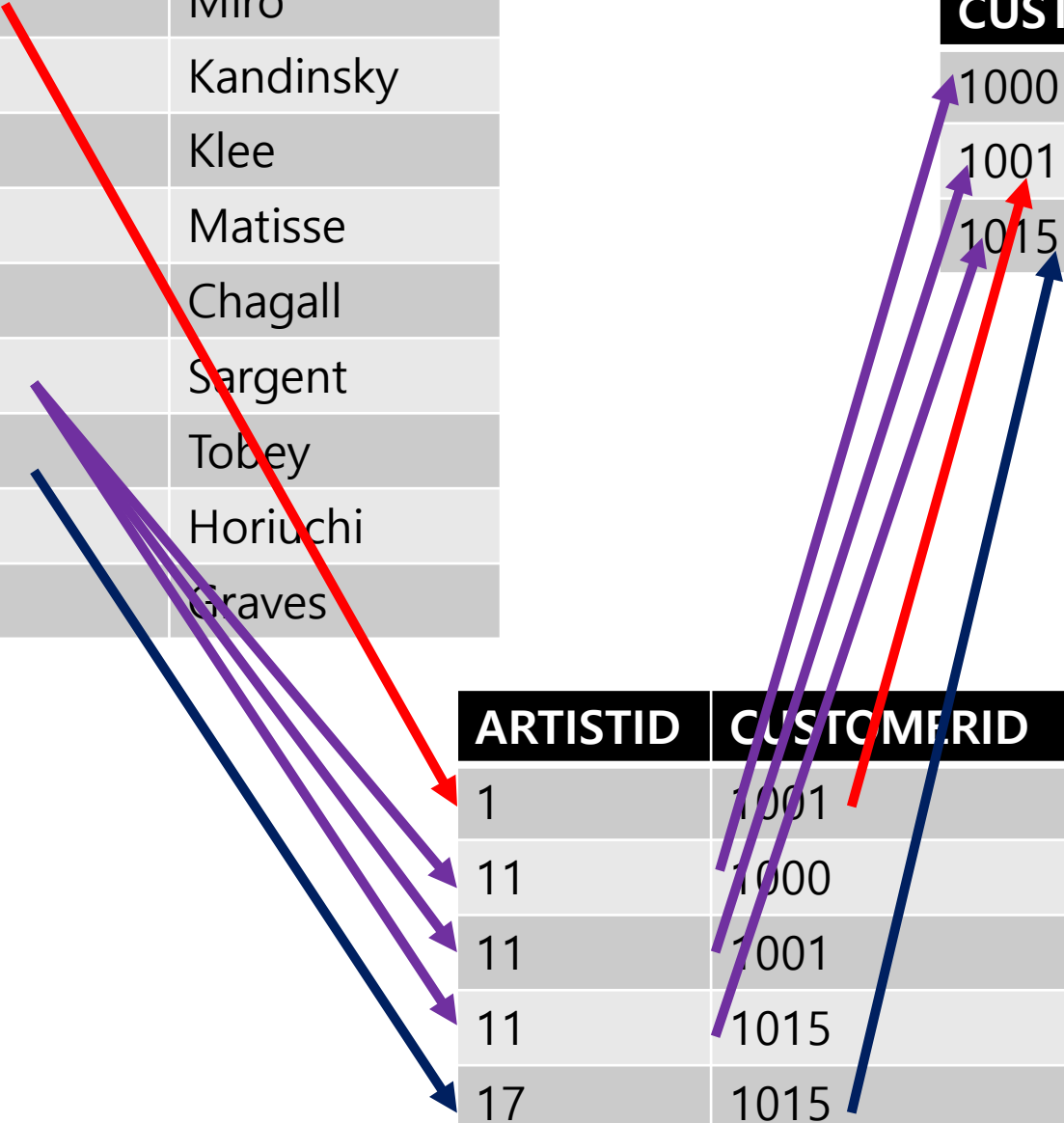
ARTISTID	LASTNAME
1	Miro
2	Kandinsky
3	Klee
4	Matisse
5	Chagall
11	Sargent
17	Tobey
18	Horiuchi
19	Graves

Customers Table

CUSTOMERID	LASTNAME
1000	Janes
1001	Smith
1015	Twilight

ARTISTID	CUSTOMERID
1	1001
11	1000
11	1001
11	1015
17	1015

Customers\_artists\_int Table





## NOT EXISTS ( 결과적으로 artistid = 11 )

```
SQL> SELECT * FROM CUSTOMERS C
2  WHERE NOT EXISTS
3      (SELECT *
4        FROM CUSTOMERS_ARTISTS_INT A
5        WHERE C.CustomerID = A.CustomerID
6        AND ArtistID = 11);
```

선택된 레코드가 없습니다.

```
SQL> SELECT * FROM CUSTOMERS C
2  WHERE NOT EXISTS
3      (SELECT *
4        FROM CUSTOMERS_ARTISTS_INT A
5        WHERE C.CustomerID = A.CustomerID
6        AND ArtistID = 1);
```

CUSTOMERID	LASTNAME	FIRSTNAME
1000	Janes	Jeffrey
1015	Twilight	Tiffany

```
SQL> SELECT * FROM CUSTOMERS C
2  WHERE NOT EXISTS
3      (SELECT *
4        FROM CUSTOMERS_ARTISTS_INT A
5        WHERE C.CustomerID = A.CustomerID
6        AND ArtistID = 17);
```

CUSTOMERID	LASTNAME	FIRSTNAME
1000	Janes	Jeffrey
1001	Smith	David

## ■ 모든 CUSTOMERS가 관심을 가지고 있는 작가를 검색

```
SELECT A.FirstName, A.LastName  
FROM ARTISTS A  
WHERE NOT EXISTS  
  ((SELECT C.CustomerID  
    FROM CUSTOMERS C)  
  MINUS  
  (SELECT CAI.CustomerID  
    FROM CUSTOMERS_ARTISTS_INT CAI  
   WHERE A.ArtistID = CAI.ArtistID));
```

FIRSTNAME	LASTNAME
John Singer	Sargent

## ■ CUSTOMERS → CUSTOMERS\_VERSION2

```
SQL> CREATE TABLE CUSTOMERS_VERSION2(  
2      CustomerID NUMBER(38) PRIMARY KEY,  
3      LastName CHAR(25) NOT NULL,  
4      FirstName CHAR(25) NOT NULL);
```

테이블이 생성되었습니다.



```
SQL> INSERT INTO CUSTOMERS_VERSION2 SELECT * FROM CUSTOMERS;
```

3 행이 생성되었습니다.

```
SQL> SELECT * FROM CUSTOMERS_VERSION2;
```

CUSTOMERID	LASTNAME	FIRSTNAME
1000	Janes	Jeffrey
1001	Smith	David
1015	Twilight	Tiffany

②



```
SQL> DROP TABLE CUSTOMERS;
```

테이블이 삭제되었습니다.

## ■ CUSTOMERS → CUSTOMERS\_VERSION2

```
SQL> SELECT * FROM CUSTOMERS;
```

CUSTOMERID	LASTNAME	FIRSTNAME
1000	Janes	Jeffrey
1001	Smith	David
1015	Twilight	Tiffany



```
SQL> CREATE TABLE CUSTOMERS_VERSION2 AS SELECT * FROM CUSTOMERS;
```

테이블이 생성되었습니다.

```
SQL> SELECT * FROM CUSTOMERS_VERSION2;
```

CUSTOMERID	LASTNAME	FIRSTNAME
1000	Janes	Jeffrey
1001	Smith	David
1015	Twilight	Tiffany

②



```
SQL> DROP TABLE CUSTOMERS;
```

테이블이 삭제되었습니다.

## ■ CUSTOMERS → CUSTOMERS\_VERSION2

- alter table <테이블명> rename to <테이블명>

```
SQL> DESC CUSTOMERS;
```

이름	널?	유형
CUSTOMERID	NOT NULL	NUMBER(38)
LASTNAME	NOT NULL	CHAR(25)
FIRSTNAME	NOT NULL	CHAR(25)



```
SQL> ALTER TABLE CUSTOMERS RENAME TO CUSTOMERS_VERSION2;
```

테이블이 변경되었습니다.

```
SQL> DESC CUSTOMERS_VERSION2;
```

이름	널?	유형
CUSTOMERID	NOT NULL	NUMBER(38)
LASTNAME	NOT NULL	CHAR(25)
FIRSTNAME	NOT NULL	CHAR(25)

## ■ CUSTOMERS 테이블에 ADDRESS 컬럼 추가

- alter table <테이블명> add <컬럼명> <타입> <조건>

```
SQL> DESC CUSTOMERS;
```

이름	널?	유형
CUSTOMERID	NOT NULL	NUMBER(38)
LASTNAME	NOT NULL	CHAR(25)
FIRSTNAME	NOT NULL	CHAR(25)



```
SQL> ALTER TABLE CUSTOMERS ADD Address VARCHAR(100) DEFAULT 'Unknown';
```

테이블이 변경되었습니다.

```
SQL> DESC CUSTOMERS;
```

이름	널?	유형
CUSTOMERID	NOT NULL	NUMBER(38)
LASTNAME	NOT NULL	CHAR(25)
FIRSTNAME	NOT NULL	CHAR(25)
ADDRESS		VARCHAR2(100)

## ■ CUSTOMERS 테이블에 ADDRESS 컬럼 타입 변경

- alter table <테이블명> modify (<컬럼명> <타입> <조건>)

```
SQL> DESC CUSTOMERS;
```

이름	널?	유형
CUSTOMERID	NOT NULL	NUMBER(38)
LASTNAME	NOT NULL	CHAR(25)
FIRSTNAME	NOT NULL	CHAR(25)
ADDRESS		VARCHAR2(100)



```
SQL> ALTER TABLE CUSTOMERS MODIFY (Address CHAR(50) NOT NULL);
```

테이블이 변경되었습니다.

```
SQL> DESC CUSTOMERS;
```

이름	널?	유형
CUSTOMERID	NOT NULL	NUMBER(38)
LASTNAME	NOT NULL	CHAR(25)
FIRSTNAME	NOT NULL	CHAR(25)
ADDRESS	NOT NULL	CHAR(50)

## ■ CUSTOMERS 테이블에 ADDRESS 컬럼 삭제

- alter table <테이블명> drop column <컬럼명>

```
SQL> DESC CUSTOMERS;
```

이름	널?	유형
CUSTOMERID	NOT NULL	NUMBER(38)
LASTNAME	NOT NULL	CHAR(25)
FIRSTNAME	NOT NULL	CHAR(25)
ADDRESS	NOT NULL	CHAR(50)



```
SQL> ALTER TABLE CUSTOMERS DROP COLUMN Address;
```

테이블이 변경되었습니다.

```
SQL> DESC CUSTOMERS;
```

이름	널?	유형
CUSTOMERID	NOT NULL	NUMBER(38)
LASTNAME	NOT NULL	CHAR(25)
FIRSTNAME	NOT NULL	CHAR(25)



## ■ ARTISTS 테이블 DateOfBirth 컬럼에 제약 추가

- alter table <테이블명> add constraint <제약조건명> <제약 조건>

```
SQL> ALTER TABLE ARTISTS ADD CONSTRAINT ARTISTS_DATA  
2 CHECK (DateOfBirth > 1800 AND DateOfBirth < 2100);
```

테이블이 변경되었습니다.



```
SQL> INSERT INTO ARTISTS VALUES(10000, 'XXX', 'XXX', 'XXX', 1700, 1900);  
INSERT INTO ARTISTS VALUES(10000, 'XXX', 'XXX', 'XXX', 1700, 1900)
```

\*

1행에 오류:

ORA-02290: 체크 제약조건 (DBDG.ARTISTS\_DATA)이 위반되었습니다

## ■ ARTISTS 테이블 DateOfBirth 컬럼에 제약 삭제

- alter table <테이블명> drop constraint <제약조건명>

```
SQL> ALTER TABLE ARTISTS DROP CONSTRAINT ARTISTS_DATA;
```

테이블이 변경되었습니다.



```
SQL> INSERT INTO ARTISTS VALUES(10000, 'XXX', 'XXX', 'XXX', 1700, 1900);
```

1 개의 행이 만들어졌습니다.

Q1> **ARTISTS**의 **Nationality**를 Spanish, Russian, German, French, United States만 입력 가능하도록 **CHECK 제약조건**을 추가하시오.

<CHECK 제약조건 작성 후에 실행(오류 발생 확인)>

```
insert into artists(artistid, lastname, firstname, nationality,  
dateofbirth, datedeceased) values(10440, 'Miro3', 'Joan3',  
'Japan', 1900, 1983);
```



1행에 오류:

ORA-02290: 체크 제약조건(ST2016127059.ARTIST\_NATIONALITY)이 위배되었습니다

Q2> 다음 중 **ARTISTS** 테이블에서 **Nationality** 컬럼의 속성을 **NULL** → **NOT NULL**로 수정하여라.

```
SQL> DESC ARTISTS;
```

이름	널?	유형
ARTISTID	NOT NULL	NUMBER(38)
LASTNAME	NOT NULL	CHAR(25)
FIRSTNAME	NOT NULL	CHAR(25)
NATIONALITY		CHAR(30)
DATEOFBIRTH		NUMBER(4)
DATEDECEASED		NUMBER(4)



```
SQL> DESC ARTISTS;
```

이름	널?	유형
ARTISTID	NOT NULL	NUMBER(38)
LASTNAME	NOT NULL	CHAR(25)
FIRSTNAME	NOT NULL	CHAR(25)
NATIONALITY	NOT NULL	CHAR(30)
DATEOFBIRTH		NUMBER(4)
DATEDECEASED		NUMBER(4)

Q3> 모든 배를 예약한 적이 있는 뱃사람의 이름과 나이를 출력하시오. (**NOT EXISTS**와 **MINUS** 사용)

Q4> **EMPLOYEES** 테이블의 이름을 **EMPLOYEES\_V2**로 변경하고, 컬럼 **Budget**을 삭제하고 **Age**를 추가하시오.

- 제출 방식 : E-Class를 통하여 제출
- 제출 내용 : spool file(Oracle에서 실행한 로그 파일)
- 제출 형식 : 학번\_이름\_주차
  - Ex) 학번\_홍길동\_7주차.zip
- 제출 기한 : 수업 시작 시간으로 부터 24시간 이내 제출
  - 제출 기한 위반 시 감점 기준
    - 지각 제출 시 과제 점수에서 40% 감점
    - 1일 초과 당 20% 추가 감점 (단, 4일 이후 제출 불가)