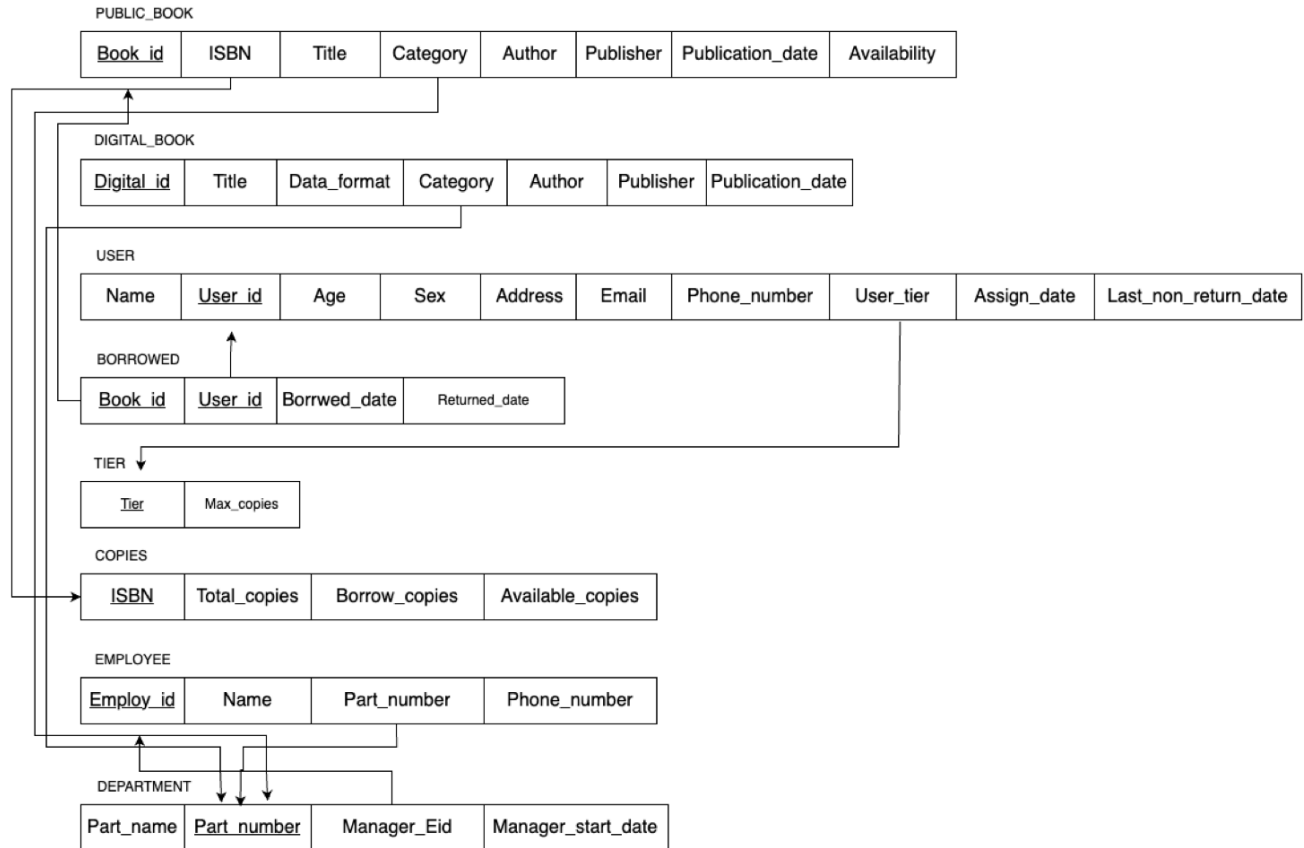


# Proj 2

## FD & NF

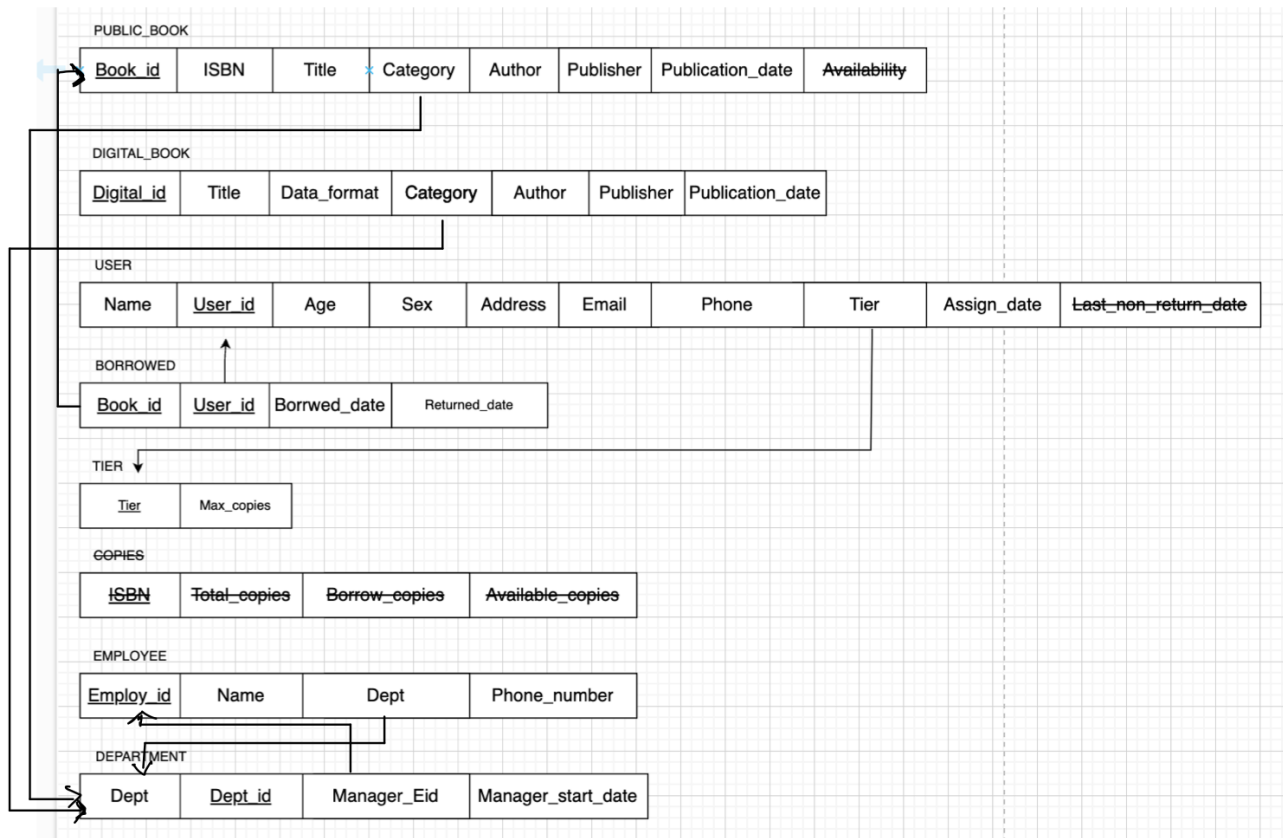
프로젝트 1에서의 Relational Model은 아래와 같았습니다.



여기서 Available와 Last\_non\_return\_date의 경우 완전함수종속을 지키지 않아, 2NF를 어기기 때문에 삭제하였고, python 파일로 생성한 데이터형태에 맞게 Part\_name -> Dept, Part\_number -> Dept\_id로 이름이 바뀌었고, 이어서 Category는 String으로 Part\_number가 아닌 Dept를 참조하도록 수정했습니다. 마찬가지로 FK에 맞게 Digital\_book의 Category도 Dept를 참조합니다.

그리고 COPIES의 경우 BORROWED에 새 대출/반납기록이 추가되면 매번 수정되어야 합니다. 또한 COPIES의 Borrow\_copies와 Available\_copies는 COPIES의 PK에 종속되는 것이 아닌 다른 BORROWED에 종속되기 때문에 3NF를 위반하고, 다른 테이블과의 JOIN으로 생성할 수 있는 테이블이기에 삭제했습니다.

아래는 수정된 Relational Model입니다.



**CREATE**

```

mysql> CREATE TABLE PUBLIC_BOOK
  -> (Book_id VARCHAR(18) NOT NULL,
  -> ISBN VARCHAR(14) NOT NULL,
  -> Title VARCHAR(50) NOT NULL,
  -> Category VARCHAR(30) NOT NULL,
  -> Author VARCHAR(30) NOT NULL,
  -> Publisher VARCHAR(50) NOT NULL,
  -> Publication_date DATE,
  -> PRIMARY KEY (Book_id));
Query OK, 0 rows affected (0.07 sec)

mysql> CREATE TABLE DIGITAL_BOOK
  -> (Digital_id VARCHAR(6) NOT NULL,
  -> Title VARCHAR(50) NOT NULL,
  -> Data_format VARCHAR(5) NOT NULL,
  -> Category VARCHAR(10) NOT NULL,
  -> Author VARCHAR(30) NOT NULL,
  -> Publisher VARCHAR(50) NOT NULL,
  -> Publication_date DATE,
  -> PRIMARY KEY (Digital_id));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE USER
  -> (User_id VARCHAR(5) NOT NULL,
  -> Name VARCHAR(30) NOT NULL,
  -> Gender VARCHAR(6) NOT NULL,
  -> Address VARCHAR(10) NOT NULL,
  -> Age INT NOT NULL,
  -> Email VARCHAR(50),
  -> Phone VARCHAR(13) NOT NULL,
  -> Tier VARCHAR(10) NOT NULL,
  -> Assign_date DATE NOT NULL,
  -> Last_non_return_date DATE NOT NULL,
  -> PRIMARY KEY (User_id));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE BORROWED
  -> (Book_id VARCHAR(18) NOT NULL,
  -> User_id VARCHAR(5) NOT NULL,
  -> Borrowed_date DATE,
  -> Returned_date DATE,
  -> PRIMARY KEY (Book_id, User_id));
Query OK, 0 rows affected (0.01 sec)

```

```

mysql> CREATE TABLE TIER
  -> (Tier VARCHAR(10) NOT NULL,
  -> Max_copies INT NOT NULL,
  -> PRIMARY KEY (Tier));
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE EMPLOYEE
  -> (Employ_id VARCHAR(5) NOT NULL,
  -> Name VARCHAR(30) NOT NULL,
  -> Dept_id INT NOT NULL,
  -> Phone VARCHAR(13) NOT NULL,
  -> PRIMARY KEY (Employ_id));
Query OK, 0 rows affected (0.02 sec)

mysql> CREATE TABLE DEPARTMENT
  -> (Dept VARCHAR(30) NOT NULL,
  -> Dept_id INT NOT NULL,
  -> Manager_eid VARCHAR(5) NOT NULL,
  -> Manager_start_date DATE NOT NULL,
  -> PRIMARY KEY (Dept_id));
Query OK, 0 rows affected (0.02 sec)

```

우선 Primary Key만 설정하고 테이블을 생성한 후 Foreign key를 추가하는 방법으로 생성했습니다.

```

mysql> ALTER TABLE DEPARTMENT ADD INDEX (Dept);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE PUBLIC_BOOK
  -> ADD CONSTRAINT fk_category FOREIGN KEY (Category) REFERENCES DEPARTMENT(Dept);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE DIGITAL_BOOK
  -> ADD CONSTRAINT fk_category_digital FOREIGN KEY (Category) REFERENCES DEPARTMENT(Dept);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE TIER ADD INDEX (Tier);
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE USER ADD CONSTRAINT fk_tier FOREIGN KEY (Tier) REFERENCES TIER(Tier);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE USER ADD INDEX (User_id);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE EMPLOYEE ADD INDEX (Employ_id);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE EMPLOYEE
  -> ADD CONSTRAINT fk_dept_id FOREIGN KEY (Dept_id) REFERENCES DEPARTMENT(Dept_id);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE DEPARTMENT
  -> ADD CONSTRAINT fk_manager FOREIGN KEY (Manager_eid) REFERENCES EMPLOYEE(Employ_id);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

```

이후 생성한 Table에 Foreign Key를 추가했습니다. 이후 python으로 생성한 tsv 데이터를 load했습니다. 로드하던 중, 아래와 같이 Title과 Publisher의 길이가 초과되는 것이 있어, 에러가 났습니다. 아래와 같이 스키마를 수정해 다시 로드했습니다.

```
mysql> SHOW WARNINGS LIMIT 10;
```

Level	Code	Message
Warning	1265	Data truncated for column 'Title' at row 2
Warning	1265	Data truncated for column 'Publisher' at row 4
Warning	1265	Data truncated for column 'Title' at row 12
Warning	1265	Data truncated for column 'Title' at row 25
Warning	1265	Data truncated for column 'Title' at row 33
Warning	1265	Data truncated for column 'Title' at row 37
Warning	1265	Data truncated for column 'Title' at row 45
Warning	1265	Data truncated for column 'Title' at row 47
Warning	1265	Data truncated for column 'Title' at row 48
Warning	1265	Data truncated for column 'Title' at row 50

10 rows in set (0.00 sec)

```
mysql> ALTER TABLE PUBLIC_BOOK MODIFY COLUMN Title VARCHAR(100);
```

Query OK, 16983 rows affected (0.13 sec)

Records: 16983 Duplicates: 0 Warnings: 0

```
mysql> ALTER TABLE DIGITAL_BOOK MODIFY COLUMN Title VARCHAR(100);
```

Query OK, 0 rows affected (0.01 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> ALTER TABLE PUBLIC_BOOK MODIFY COLUMN Publisher VARCHAR(100);
```

Query OK, 16983 rows affected (0.14 sec)

Records: 16983 Duplicates: 0 Warnings: 0

```
mysql> ALTER TABLE DIGITAL_BOOK MODIFY COLUMN Publisher VARCHAR(100);
```

Query OK, 0 rows affected (0.02 sec)

Records: 0 Duplicates: 0 Warnings: 0

```
mysql> TRUNCATE TABLE PUBLIC_BOOK;
```

Query OK, 0 rows affected (0.02 sec)

```
mysql> LOAD DATA LOCAL INFILE '/Users/jeong-yonghun/Desktop/3-1 과제 /DataBase/Library/book_new.tsv'
```

-> INTO TABLE PUBLIC\_BOOK

-> FIELDS TERMINATED BY '\t'

-> LINES TERMINATED BY '\n'

-> IGNORE 1 LINES

-> (Book\_id, ISBN, Title, Category, Author, Publisher, Publication\_date);

Query OK, 16983 rows affected, 468 warnings (0.19 sec)

Records: 16983 Deleted: 0 Skipped: 0 Warnings: 468

```
mysql> LOAD DATA LOCAL INFILE '/Users/jeong-yonghun/Desktop/3-1 과제/DataBase/Library/book_digital.tsv'
-> INTO TABLE DIGITAL_BOOK
-> FIELDS TERMINATED BY '\t'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES
-> (Digital_id, Title, Data_format, Category, Author, Publisher, Publication_date);
Query OK, 1021 rows affected, 26 warnings (0.02 sec)
Records: 1021 Deleted: 0 Skipped: 0 Warnings: 26
```

```
mysql> SELECT * FROM DIGITAL_BOOK
-> LIMIT 5;
```

Digital_id	Title	Publication_date	Data_format	Category	Author	Publisher
000001	(후니의 쉽게 쓴) 시스코 네트워크	2018	JPEG	Digital	진강훈	BM 성안당
000002	Java 2 -	2003	MOBI	Digital	Schildt, Herbert	대광서림
000003	컴파일러	2009	MOBI	Digital	Alfred V. Aho	피어슨에듀케이
000004	컴파일러	2009	MOBI	Digital	Alfred V. Aho	피어슨에듀케이
000005	모든 정부는 거짓말을 한다	2012	EPUB	Digital	MacPherson, Myra	문학동네

5 rows in set (0.00 sec)

```
mysql> LOAD DATA LOCAL INFILE '/Users/jeong-yonghun/Desktop/3-1 과제/DataBase/Library/student.tsv'
-> INTO TABLE USER
-> FIELDS TERMINATED BY '\t'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES
-> (User_id, Name, Gender, Address, Age, Email, Phone, Tier, Assign_date, Last_non_return_date);
Query OK, 1000 rows affected (0.02 sec)
Records: 1000 Deleted: 0 Skipped: 0 Warnings: 0
```

```
mysql> SELECT * FROM USER
-> LIMIT 5;
```

User_id	Name	Gender	Address	Age	Email	Phone	Tier	Assign_date	Last_non_return_date
10000-15	Sarah	Female	Daejeon	23	sarah_01s@ajou.ac.kr	010-2514-1099	Gold	2023-02-21	2024-05
10001-11	Amanda	Female	Incheon	22	amanda_02a@ajou.ac.kr	010-2778-6548	Gold	2022-09-27	2024-01
10002-07	Elizabeth	Female	Ulsan	20	elizabeth_04e@ajou.ac.kr	010-4147-8170	Silver	2022-07-31	2024-04
10003-04	Deborah	Female	Daejeon	20	deborah_04d@ajou.ac.kr	010-9518-7622	Iron	2022-12-31	2024-04
10004-13	Matthew	Male	Seoul	20	matthew_04m@ajou.ac.kr	010-2778-3391	Iron	2022-07-22	2024-01

5 rows in set (0.00 sec)

```
mysql> LOAD DATA LOCAL INFILE '/Users/jeong-yonghun/Desktop/3-1 과제 /DataBase/Library/logs.tsv'
-> INTO TABLE temp_borrowed
-> FIELDS TERMINATED BY '\t'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES
-> (Book_id, User_id, @Borrowed_date, @Returned_date)
-> SET
->     Borrowed_date = STR_TO_DATE(NULLIF(@Borrowed_date, ''), '%Y-%m-%d %H:%i:%s'),
->     Returned_date = STR_TO_DATE(NULLIF(@Returned_date, ''), '%Y-%m-%d %H:%i:%s');
```

```
mysql> LOAD DATA LOCAL INFILE '/Users/jeong-yonghun/Desktop/3-1 과제 /DataBase/Library/tier.tsv'
-> INTO TABLE TIER
-> FIELDS TERMINATED BY '\t'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES
-> (Tier, Max_copies);
Query OK, 4 rows affected (0.00 sec)
Records: 4 Deleted: 0 Skipped: 0 Warnings: 0
```

```
mysql> LOAD DATA LOCAL INFILE '/Users/jeong-yonghun/Desktop/3-1 과제 /DataBase/Library/employ.tsv'
-> INTO TABLE EMPLOYEE
-> FIELDS TERMINATED BY '\t'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES
-> (Employ_id, Name, Dept_id, Phone);
Query OK, 1001 rows affected (0.02 sec)
Records: 1001 Deleted: 0 Skipped: 0 Warnings: 0
```

```
mysql> LOAD DATA LOCAL INFILE '/Users/jeong-yonghun/Desktop/3-1 과제 /DataBase/Library/dept.tsv'
-> INTO TABLE DEPARTMENT
-> FIELDS TERMINATED BY '\t'
-> LINES TERMINATED BY '\n'
-> IGNORE 1 LINES
-> (Dept, Dept_id, Manager_eid, Manager_start_date);
Query OK, 11 rows affected (0.00 sec)
Records: 11 Deleted: 0 Skipped: 0 Warnings: 0
```

## Constraints

1. Return item before borrowing : 아래와 같이 Constraint 쿼리를 작성했으나 생성된 데이터 자체에서 constraint를 지키지 않는 경우가 있어 예러가 났고 select로 확인해본 결과, 실제 그런 데이터들이

존재해 해당 데이터를 DELETE한 뒤, constraint를 추가했습니다.

```
mysql> ALTER TABLE BORROWED
-> ADD CONSTRAINT check_return_before_borrow
-> CHECK (Returned_date IS NULL OR Returned_date >= Borrowed_date);
ERROR 3819 (HY000): Check constraint 'check_return_before_borrow' is violated.
mysql> SELECT * FROM BORROWED
-> WHERE Returned_date IS NOT NULL AND Returned_date < Borrowed_date
-> LIMIT 10;
```

Book_id	User_id	Borrowed_date	Returned_date
9780007427925001	10045	2021-12-30 14:55:01	2021-08-17 10:23:36
9780060087647001	10306	2021-12-30 13:39:07	2021-08-29 11:55:30
9780060891756001	10687	2021-12-24 14:23:33	2021-09-27 11:04:12
9780061472794001	10979	2021-12-30 17:13:41	2021-10-11 09:24:45
9780062942937001	10610	2021-12-24 14:33:46	2021-08-11 10:44:35
9780070151451001	10018	2021-12-30 13:55:41	2021-05-28 11:18:30
9780070166783001	10024	2021-12-26 17:03:00	2021-11-04 11:26:54
9780070592926002	10422	2021-12-30 15:06:14	2021-07-02 10:13:42
9780070707313002	10372	2021-12-24 16:09:23	2021-09-24 11:16:11
9780071235259001	10386	2021-12-29 15:37:49	2021-07-26 10:11:29

```
10 rows in set (0.00 sec)

mysql> DELETE FROM BORROWED
-> WHERE Returned_date IS NOT NULL AND Returned_date < Borrowed_date;
Query OK, 3855 rows affected (0.13 sec)

mysql> ALTER TABLE BORROWED
-> ADD CONSTRAINT check_return_before_borrow
-> CHECK (Returned_date IS NULL OR Returned_date >= Borrowed_date);
Query OK, 212017 rows affected (0.95 sec)
Records: 212017 Duplicates: 0 Warnings: 0
```



## 2. Borrow item which is already borrowed

```
mysql> ALTER TABLE BORROWED
  -> ADD CONSTRAINT unique_book_borrowed
  -> UNIQUE (Book_id, Borrowed_date);
Query OK, 0 rows affected (0.42 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> SHOW CREATE TABLE BORROWED;
+-----+-----+
| Table | Create Table |
+-----+-----+
| BORROWED | CREATE TABLE `BORROWED` (
  `Book_id` varchar(18) NOT NULL,
  `User_id` varchar(5) NOT NULL,
  `Borrowed_date` datetime DEFAULT NULL,
  `Returned_date` datetime DEFAULT NULL,
  PRIMARY KEY (`Book_id`,`User_id`),
  UNIQUE KEY `unique_book_borrowed` (`Book_id`,`Borrowed_date`),
  KEY `fk_user_id` (`User_id`),
  CONSTRAINT `fk_user_id` FOREIGN KEY (`User_id`) REFERENCES `USER` (`User_id`),
  CONSTRAINT `check_return_before_borrow` CHECK (((`Returned_date` is null) or (`Returned_date` >= `Borrowed_date`)))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+-----+-----+
1 row in set (0.00 sec)
```

+Constraint를 진행하던 중, 위에서 언급한 NF 관련 Drop

```
mysql> ALTER TABLE USER
-> DROP COLUMN Last_non_return_date;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESCRIBE USER;
```

Field	Type	Null	Key	Default	Extra
User_id	varchar(5)	NO	PRI	NULL	
Name	varchar(30)	NO		NULL	
Gender	varchar(6)	NO		NULL	
Address	varchar(10)	NO		NULL	
Age	int	NO		NULL	
Email	varchar(50)	YES		NULL	
Phone	varchar(13)	NO		NULL	
Tier	varchar(10)	NO	MUL	NULL	
Assign_date	date	NO		NULL	

9 rows in set (0.00 sec)

3. 제가 정의한 Book\_id는 ISBN에 3자리 식별자를 추가한 형태로, Book\_id의 앞의 13자리와 ISBN은 같아야 합니다. 해당 Constraints

```
mysql> ALTER TABLE PUBLIC_BOOK
-> ADD CONSTRAINT chk_book_id_isbn
-> CHECK (SUBSTR(Book_id, 1, 13) = SUBSTR(ISBN, 1, 13));
Query OK, 16983 rows affected (0.17 sec)
Records: 16983 Duplicates: 0 Warnings: 0
```

## DELETE

1. Delete book : 가장 번호가 큰 유저를 삭제했습니다.

```
mysql> SELECT MAX(User_id) AS max_user_id
-> FROM BORROWED
-> ORDER BY User_id DESC
-> LIMIT 1;
+-----+
| max_user_id |
+-----+
| 10999       |
+-----+
1 row in set (0.00 sec)

mysql> SELECT Book_id, User_id FROM BORROWED WHERE User_id LIKE '10999'
-> LIMIT 5;
+-----+-----+
| Book_id          | User_id |
+-----+-----+
| 9780128002070001 | 10999   |
| 9780136073734001 | 10999   |
| 9780201423006001 | 10999   |
| 9780321544285001 | 10999   |
| 9780321714114001 | 10999   |
+-----+-----+
5 rows in set (0.00 sec)

mysql> DELETE FROM BORROWED
-> WHERE User_id = '10999' AND Book_id = '9780128002070001';
Query OK, 1 row affected (0.00 sec)
```

2. Delete borrow history (user\_id, book\_id) : 가장 미반납기간이 긴 유저의 대출기록을 하나 지워줬습니다.

```
mysql> SELECT User_id, COUNT(*) AS non_returned_count
-> FROM BORROWED
-> WHERE Returned_date IS NULL
-> GROUP BY User_id
-> ORDER BY non_returned_count DESC
-> LIMIT 1;
```

```
+-----+-----+
| User_id | non_returned_count |
+-----+-----+
| 10393   | 7                  |
+-----+-----+
```

1 row in set (0.22 sec)

```
mysql> SELECT Book_id, User_id, Returned_date FROM BORROWED
-> WHERE User_id LIKE '10393' AND Returned_date IS NULL;
```

```
+-----+-----+-----+
| Book_id          | User_id | Returned_date |
+-----+-----+-----+
| 9788925588681001 | 10393   | NULL          |
| 9788931423112001 | 10393   | NULL          |
| 9788931461329001 | 10393   | NULL          |
| 9788965402879001 | 10393   | NULL          |
| 9788995890967001 | 10393   | NULL          |
| 9791189620844001 | 10393   | NULL          |
| 9791190764131001 | 10393   | NULL          |
+-----+-----+-----+
```

7 rows in set (0.01 sec)

```
mysql> DELETE FROM BORROWED
-> WHERE User_id = '10393' AND Book_id = '9788925588681001';
Query OK, 1 row affected (0.01 sec)
```

3. 제목이 30보다 긴 책을 한권 찾아 지웠습니다.

```
mysql> SELECT Book_id FROM PUBLIC_BOOK
-> WHERE LENGTH(Title) >= 30
-> LIMIT 5;

+-----+
| Book_id          |
+-----+
| 9780060891756001 |
| 9780060955366001 |
| 9780061132384001 |
| 9780061431616001 |
| 9780061472794001 |
+-----+
5 rows in set (0.01 sec)

mysql> DELETE FROM PUBLIC_BOOK
-> WHERE Book_id = '9780060891756001';
Query OK, 1 row affected (0.00 sec)
```

## UPDATE

1. Update user info : 광복절을 맞이하여 Iron Tier이고 등록일이 오래 된 유저 5명의 티어를 한 단계 올려줍니다.

```
mysql> UPDATE USER
-> SET Tier = 'Bronze'
-> WHERE Tier = 'Iron'
-> ORDER BY Assign_date
-> LIMIT 5;
Query OK, 5 rows affected (0.01 sec)
Rows matched: 5  Changed: 5  Warnings: 0
```

2. Update item info : JPEG 타입의 Digital\_book이 이제 모두 PDF로 보관하기로 하여, 모든 JPEG을 PDF 로 바꿨습니다.

```
mysql> UPDATE DIGITAL_BOOK  
      -> SET Data_format = 'PDF'  
      -> WHERE Data_format = 'JPEG';  
Query OK, 252 rows affected (0.01 sec)  
Rows matched: 252  Changed: 252  Warnings: 0
```

3. 대출 하려는 사람이 많아, 모든 티어의 사용자의 대출가능권수를 3배 늘리기로 결정해, 업데이트했습니다.

```
mysql> UPDATE TIER  
      -> SET Max_copies = Max_copies * 3;  
Query OK, 3 rows affected (0.00 sec)  
Rows matched: 4  Changed: 3  Warnings: 0
```

## Retrieval

## 1. Top-10 popular items

```
mysql> SELECT p.Title, COUNT(b.Book_id) AS Borrows
-> FROM BORROWED b
-> INNER JOIN PUBLIC_BOOK p ON b.Book_id = p.Book_id
-> WHERE b.Returned_date IS NULL
-> GROUP BY p.Title
-> ORDER BY Borrows DESC
-> LIMIT 10;
```

Title	Borrows
인공지능	3
네트워크 시뮬레이터	3
운영체제	3
메타버스	3
당신들의 천국	2
Database system concepts	2
고전 강연	2
인공지능 시스템	2
The organic chemistry of drug design and drug action	2
Discrete-event system simulation	2

10 rows in set (0.14 sec)

## 2. Top-10 active users

```
mysql> SELECT u.Name, COUNT(b.Book_id) AS Borrows
-> FROM USER u
-> INNER JOIN BORROWED b ON u.User_id = b.User_id
-> WHERE b.Returned_date IS NULL
-> GROUP BY u.Name
-> ORDER BY Borrows DESC
-> LIMIT 10;
```

Name	Borrows
Yoonseo	32
Emily	31
Joseph	27
Jimin	26
Jiyoo	25
Doyoon	24
Donald	24
Siu	23
Jason	23
Matthew	23

```
10 rows in set (0.07 sec)
```



### 3. Top-3 single book lover

```
mysql> SELECT u.Name, COUNT(DISTINCT pb.ISBN) AS UniqueISBNs
-> FROM USER u
-> INNER JOIN BORROWED b ON u.User_id = b.User_id
-> INNER JOIN PUBLIC_BOOK pb ON b.Book_id = pb.Book_id
-> WHERE b.Returned_date IS NULL
-> GROUP BY u.Name
-> ORDER BY UniqueISBNs DESC
-> LIMIT 3;

+-----+-----+
| Name   | UniqueISBNs |
+-----+-----+
| Yoonseo |          32 |
| Emily   |          31 |
| Joseph  |          27 |
+-----+-----+
3 rows in set (0.13 sec)
```

### 4. Top-3 early-bird users

```
mysql> SELECT u.Name, AVG(DATEDIFF(b.Returned_date, b.Borrowed_date)) AS AvgReturnDays
-> FROM USER u
-> INNER JOIN BORROWED b ON u.User_id = b.User_id
-> WHERE b.Returned_date IS NOT NULL
-> GROUP BY u.Name
-> ORDER BY AvgReturnDays ASC
-> LIMIT 3;

+-----+-----+
| Name   | AvgReturnDays |
+-----+-----+
| Ryan   |          3.8306 |
| Steven |          3.8502 |
| Dohyun |          3.8527 |
+-----+-----+
3 rows in set (0.31 sec)
```

## 5. Average # of books borrowed (by each dept.)

```
mysql> SELECT D.Dept, COUNT(B.Book_id) / COUNT(DISTINCT U.User_id) AS AvgBooks FROM BORROWED AS B
-> JOIN USER AS U ON B.User_id = U.User_id
-> JOIN PUBLIC_BOOK AS P ON B.Book_id = P.Book_id
-> JOIN DEPARTMENT AS D ON P.Category = D.Dept
-> GROUP BY D.Dept;
```

Dept	AvgBooks
art	19.5810
digital	18.9720
etc	19.8690
history	18.3870
language	20.1510
literature	19.9720
natural science	18.4230
Philosophy	18.5290
religion	19.0920
social science	20.0170
technological science	19.0090

11 rows in set (0.44 sec)

## 6. Average return day (by each dept.)

```
mysql> SELECT D.Dept, AVG(DATEDIFF(B.Returned_date, B.Borrowed_date)) AS AvgReturnDay FROM BORROWED AS B
-> JOIN USER AS U ON B.User_id = U.User_id
-> JOIN PUBLIC_BOOK AS P ON B.Book_id = P.Book_id
-> JOIN DEPARTMENT AS D ON P.Category = D.Dept
-> WHERE B.Returned_date IS NOT NULL
-> GROUP BY D.Dept;
```

Dept	AvgReturnDay
art	3.9867
digital	4.0476
etc	3.9841
history	3.9631
language	3.9934
literature	4.0171
natural science	4.0380
Philosophy	4.0084
religion	3.9670
social science	3.9759
technological science	3.9781

11 rows in set (0.32 sec)

## 7. The longest borrow period (by each dept.)

```
mysql> SELECT D.Dept, MAX(DATEDIFF(B.Returned_date, B.Borrowed_date)) AS LongestBorrowPeriod FROM BORROWED AS B
-> JOIN USER AS U ON B.User_id = U.User_id
-> JOIN PUBLIC_BOOK AS P ON B.Book_id = P.Book_id
-> JOIN DEPARTMENT AS D ON P.Category = D.Dept
-> WHERE B.Returned_date IS NOT NULL
-> GROUP BY D.Dept;
```

Dept	LongestBorrowPeriod
art	20
digital	20
etc	20
history	20
language	20
literature	20
natural science	20
Philosophy	20
religion	20
social science	20
technological science	20

11 rows in set (0.32 sec)

## 8. . 성별 별 사용자 수 조회

```
mysql> SELECT Gender, COUNT(*) AS StudentCount FROM USER
-> GROUP BY Gender;
```

Gender	StudentCount
Female	489
Male	511

2 rows in set (0.00 sec)

## 9. 지역 별 사용자 수 조회

```
mysql> SELECT Address, COUNT(*) AS StudentCount FROM USER  
-> GROUP BY Address;
```

```
+-----+-----+  
| Address | StudentCount |  
+-----+-----+  
| Daejeon |          132 |  
| Incheon |          127 |  
| Ulsan   |          136 |  
| Seoul   |          117 |  
| Daegu    |          133 |  
| Gwangju |          119 |  
| Busan    |          127 |  
| Suwon    |          109 |  
+-----+-----+  
8 rows in set (0.00 sec)
```

10. . 20세에서 25세 사이의 사용자가 가장 많이 빌린 책의 카테고리 조회

```
mysql> SELECT pb.Category, COUNT(*) as BorrowCount
-> FROM BORROWED b
-> JOIN USER u ON b.User_id = u.User_id
-> JOIN PUBLIC_BOOK pb ON b.Book_id = pb.Book_id
-> WHERE u.Age BETWEEN 20 AND 25
-> GROUP BY pb.Category
-> ORDER BY BorrowCount DESC;
```

```
+-----+-----+
| Category                | BorrowCount |
+-----+-----+
| language                | 20151       |
| social science          | 20017       |
| literature              | 19972       |
| etc                     | 19869       |
| art                    | 19581       |
| religion                | 19092       |
| technological science   | 19009       |
| digital                 | 18972       |
| Philosophy              | 18529       |
| natural science         | 18423       |
| history                 | 18387       |
+-----+-----+
11 rows in set (0.21 sec)
```