Database Systems Laboratory Homework 3

정보통신공학과

2017112200 신현호

2017112207 이건우

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9ef898fc-c679-4de6-8a40-8122fb45f8c1/lab-hw3-2021.pdf

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/108e4d1f-9d32-4f64-a969-5ee05588ac44/Database_Systems_Laboratory_Homework_3.pdf

Nov. 10, 2021
O You MUST use the tables used in HW2. If you need to change your tables, you should submit the revised schema diagram of your tables.

Select four queries from the SQL queries in HW2. Any two queries can't belong to the same category(I - VI). All the queries should include one or more host variables.

1. show the four queries from HW2 you selected, including the host variables.

```
-- showing product_id of which price is more than 35000 won and quantity is over two
select P.product_id
from product as P
where P.price > 35000 and product_id in (select I.product_id
                                         from item_cart as I
                                         where I.quantity > 2);
```

	product_id
▶	74022
*	NULL

```
-- showing user's name whose gender is male and who has a product in his item cart and product's id is '372'
select U.name
from user as U
where U.gender = 'M' and U.user_id = (select I.user_id
                                      from item_cart as I
                                      where I.product_id = '372');
```

	name
▶	호현호

```
-- if there is no over-priced product, then show all the products
select *
from product
where not exists
  (select product_id
   from product
   where price > 1000000000);
```

	product_id	product_name	price	category	product_desc	sku
▶	372	black wallet	110000	wallets	for business men	W384248
	34892	students bag	44000	bags	for students	B939309
	45924	red shoes	33000	shoes	for women	K898290
	74022	micky mouse hoodie	66000	hoodies	disney merch	H383024
	88392	khaki shorts	22000	shorts	military style	S839329

```
-- if there's on sale more than 50%, then show all the products
select *
from product
where exists
  (select *
   from discount
   where discount_percent > 50);
```

product_id	product_name	price	category	product_desc	sku
372	black wallet	110000	wallets	for business men	W384248
34892	students bag	44000	bags	for students	B939309
45924	red shoes	33000	shoes	for women	K898290
74022	micky mouse hoodie	66000	hoodies	disney merch	H383024
88392	khaki shorts	22000	shorts	military style	S839329

2. write a C (or Python etc) program which can process the four SQL queries.
   You should include as many comments as possible to make it easier to be read

```
# calling module for using mysql with Python
import pymysql

# calling db
hm_db = pymysql.connect(
   user = 'root',  #user name
   passwd = '1234', #password
   host = '127.0.0.1', #host address
   db = 'hmdb', #name of db
   charset = 'utf8' #using utf8 since we use Korean
)

# creating cursor for interacting with connected DB
cursor = hm_db.cursor()

sql_01 = "select P.product_id from product as P where P.price > 35000 and product_id in (select I.product_id from item_cart as I where I.quantity > 2);"
cursor.execute(sql_01)
result_01 = cursor.fetchall()
print(result_01)

cursor = hm_db.cursor()
sql_02 = "select U.name  from user as U where U.gender = 'M' and U.user_id = (select I.user_id from item_cart as I where I.product_id = '372');"
cursor.execute(sql_02)
result_02 = cursor.fetchall()
print(result_02)

cursor = hm_db.cursor()
sql_03 = "select * from product where not exists (select product_id from product where price > 1000000000);"
cursor.execute(sql_03)
result_03 = cursor.fetchall()
print(result_03)

cursor = hm_db.cursor()
sql_04 = "select * from product where exists (select * from discount where discount_percent > 50);"
cursor.execute(sql_04)
result_04 = cursor.fetchall()
print(result_04)
```

3. show that your program works.

```
sql_01 = "select P.product_id from product as P where P.price > 35000 and product_id in (select I.product_id from item_
cursor.execute(sql_01)
result_01 = cursor.fetchall()
print(result_01)
```

```
((74022,),)
```

```
sql_02 = "select U.name  from user as U where U.gender = 'M' and U.user_id = (select I.user_id from item_cart as I where
cursor.execute(sql_02)
result_02 = cursor.fetchall()
print(result_02)
```

```
(('호현호',),)
```

```
sql_03 = "select * from product where not exists (select product_id from product where price > 1000000000);"
cursor.execute(sql_03)
result_03 = cursor.fetchall()
print(result_03)
```

```
((372, 'black_wallet', 110000, 'wallets', 'for business men', 'W384248'), (34892, 'students_bag', 44000, 'bags', 'for students', 'B939309'), (45924, 're
d_shoes', 33000, 'shoes', 'for women', 'K898290'), (74022, 'micky_mouse_hoodie', 66000, 'hoodies', 'disney merch', 'H383024'), (88392, 'khaki_shorts', 2
2000, 'shorts', 'military style', 'S839329'))
```

```
sql_04 = "select * from product where exists (select * from discount where discount_percent > 50);"
cursor.execute(sql_04)
result_04 = cursor.fetchall()
print(result_04)
```

```
((372, 'black_wallet', 110000, 'wallets', 'for business men', 'W384248'), (34892, 'students_bag', 44000, 'bags', 'for students', 'B939309'), (45924, 'red_shoes', 33000, 'shoes', 'for women', 'K898290'), (74022, 'micky_mouse_hoodie', 66000, 'hoodies', 'disney merch', 'H383024'), (88392, 'khaki_shorts', 22000, 'shorts', 'military style', 'S839329'))
```

Implement the following.

1. define a procedure which contains an aggregate function.

```
show global variables like "log_bin_trust_function_creators";
set global log_bin_trust_function_creators = 1;

-- a procedure which shows the number of users in 'User' table
drop procedure if exists sum_of_user;
delimiter &
create procedure sum_of_user()
begin
  select count(*) from user;
end &
delimiter ;

call sum_of_user();
```

count(*)
5

2. define a function which returns a value.

```
-- returns the user_id of newly inserted person
drop function if exists insert_user;
delimiter $$
create function insert_user(
    username varchar(30),
    name varchar(20),
    gender varchar(5),
    address varchar(20),
    telephone_number varchar(13)
)returns int
begin
  declare user_id int;
    select count(*) into user_id from user;
    insert into user values(user_id+1, username, name, gender,address,telephone_number);
    return user_id + 1;
end $$
delimiter ;

select insert_user('gugu', 'Jame','F','수원시 영통구','0104880778000');
```

insert_user('gugu', 'Jame','F','수원시 영통구','01...
6

```
select * from user;
```

user_id	username	n...	gender	address	telephone_numb...
1	hhshin	···	M	경기도 수원시 영통구	01000000000
2	gunnoo	···	M	서울특별시 마포구	01000000001
▶ 3	Kindess	···	F	대구광역시 달성구	01000000011
4	James98	···	M	서울특별시 중구	01000000111
5	hello02	···	M	서울특별시 성북구	01011000000
6	gugu	J...	F	수원시 영통구	0104880778000

3. write a C (or Python etc) program which calls/uses both of the procedures/functions defined in 1)-2).

```
# executing procedure
cursor = hm_db.cursor()
cursor.callproc('sum_of_user')
result_05 = cursor.fetchall()
print(result_05)

# executing function
cursor = hm_db.cursor()
func = "select insert_user(%s,%s,%s,%s,%s)"
result_06 = cursor.execute(func, ('hihi','Lisa','F','서울시 성북구','821048807780'))
result_06 = cursor.fetchall()
sql_07 = cursor.execute("select * from User;")
```

```
result_07 = cursor.fetchall()
print(result_07)
```

4. show that your program works.

```
cursor.callproc('sum_of_user')
result_05 = cursor.fetchall()
print(result_05)
```

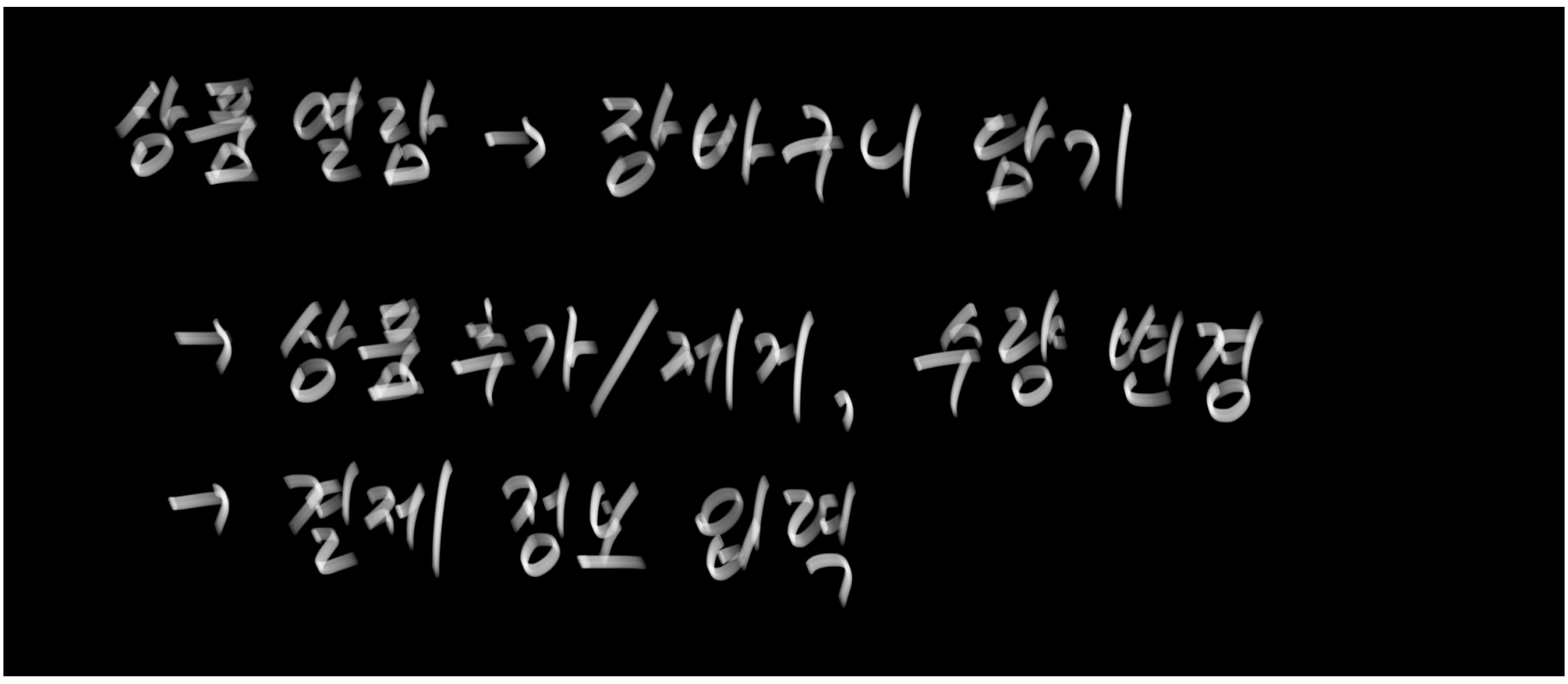
```
((6,),)
```

```
# executing function
cursor = hm_db.cursor()
func = "select insert_user(%s,%s,%s,%s,%s)"
result_06 = cursor.execute(func, ('hihi','Lisa','F','서울시 성북구','821048807780'))
result_06 = cursor.fetchall()
sql_07 = cursor.execute("select * from User;")
result_07 = cursor.fetchall()
print(result_07)
```

```
((1, 'hhshin', '신현호', 'M', '경기도 수원시 영통구', '01000000000'), (2, 'gunnoo', '이건우', 'M', '서울특별시 마포구', '01000000001'), (3, 'Kindess', '박친절', 'F', '대구광역시 달성구', '01000000011'), (4, 'James98', '호현호', 'M', '서울특별시 중구', '01000000111'), (5, 'hello02', '우건우', 'M', '서울특별시 성북구', '01011000000'), (6, 'gugu', 'Jame', 'F', '수원시 영통구', '0104880778000'), (7, 'hihi', 'Lisa', 'F', '서울시 성북구', '821048807780'))
```

The boundary of the Question 3 is flexible, and you can do as much work as you like to do. The grade will depend on the amount and quality of your work. Among the tasks you intended to do in HW 1, implement some of (if not all of, or more than) them. For each task you will implement, provide the following information in the report.

1. describe what you are going to do with the use of web pages. Show the current web pages or you can draw ones yourself.



2. implement a program using host language plus Mysql.

```
# 신상품 추가
cursor = hm_db.cursor()
sql = "insert into product values(%s,%s,%s,%s,%s,%s)"
values = (382,'blue_jacket',49000,'jackets','for men','JK102476')
cursor.execute(sql,values)
hm_db.commit()
cursor.fetchall()

#상품 열람
cursor = hm_db.cursor()
sql = "select * from product;"
cursor.execute(sql)
cursor.fetchall()

# 장바구니 담기
cursor = hm_db.cursor()
sql = "insert into item_cart values(%s, %s, %s,%s)"
values = (255,1,382,1)
cursor.execute(sql,values)
hm_db.commit()
cursor.fetchall()

# 장바구니 열람
cursor = hm_db.cursor()
```

```
sql = "select * from item_cart"
cursor.execute(sql)
cursor.fetchall()

# 상품 제거
cursor = hm_db.cursor()
sql = "delete from item_cart where product_id = %s" % (34892)
cursor.execute(sql)
cursor.fetchall

# 수량 변경
cursor = hm_db.cursor()
sql = "update table item_cart set quantity= 3 where item_cart_id = 255;"
cursor.execute(sql)
cursor.fetchall()

# order_items
cursor = hm_db.cursor()
sql = "insert into order_items values(%s,%s);"(7676,382)
cursor.execute(sql)
hm_db.commit()
cursor.fetchall()

# order_details 작성
cursor = hm_db.cursor()
sql = "insert into order_details values(%s,%s,%s,%s,%s)" % (8787, 382, 49000*3, 1, 7676)
cursor.execute(sql)
hm_db.commit()
cursor.fetchall()

# order detail 열람
cursor = hm_db.cursor()
sql = "select * from order_details"
cursor.execute(sql)
cursor.fetchall()

# payment_details 작성
cursor = hm_db.cursor()
sql = "Insert into payment_details values(3535,3,'HM','paid','KR00',8787)"
cursor.execute(sql)
hm_db.commit()
cursor.fetchall()

# payment_details 열람
cursor = hm_db.cursor()
sql = "select * from payment_details"
cursor.execute(sql)
hm_db.commit()
cursor.fetchall()
```

```
# 신상품 추가
cursor = hm_db.cursor()

sql = "insert into product values(%s,%s,%s,%s,%s,%s)"
values = (382,'blue_jacket',49000,'jackets','for men','JK102476')
cursor.execute(sql,values)
hm_db.commit()
cursor.fetchall()

#상품 열람
cursor = hm_db.cursor()
sql = "select * from product;"
cursor.execute(sql)
cursor.fetchall()
```

```
((372, 'black wallet', 100000, 'wallets', 'for business men', 'W384248'),
 (382, 'blue_jacket', 49000, 'jackets', 'for men', 'JK102476'),
 (34892, 'students bag', 40000, 'bags', 'for students', 'B939309'),
 (45924, 'red shoes', 30000, 'shoes', 'for women', 'K898290'),
 (74022, 'micky mouse hoodie', 60000, 'hoodies', 'disney merch', 'H383024'),
 (88392, 'khaki shorts', 20000, 'shorts', 'military style', 'S839329'))
```

```python
# 장바구니 담기

sql = "insert into item_cart values(%s, %s, %s,%s)"
values = (255,1,382,1)
cursor.execute(sql,values)
hm_db.commit()
cursor.fetchall()
```

```
()
```

```python
# 장바구니 열람
cursor = hm_db.cursor()

sql = "select * from item_cart"
cursor.execute(sql)
cursor.fetchall()
```

```
((200, 1, 45924, 1),
 (210, 2, 34892, 10),
 (220, 3, 88392, 1),
 (230, 2, 372, 11),
 (240, 7, 74022, 111),
 (255, 1, 382, 1))
```

```python
# 상품 제거
cursor = hm_db.cursor()
sql = "delete from item_cart where product_id = %s" % (34892)
cursor.execute(sql)
cursor.fetchall
```

```
<bound method Cursor.fetchall of <pymysql.cursors.Cursor object at 0x7fe8a028e970>>
```

```python
# 장바구니 열람
cursor = hm_db.cursor()

sql = "select * from item_cart"
cursor.execute(sql)
cursor.fetchall()
```

```
((200, 1, 45924, 1),
 (220, 3, 88392, 1),
 (230, 2, 372, 11),
 (240, 7, 74022, 111),
 (255, 1, 382, 1))
```

```python
# 수량 변경
cursor = hm_db.cursor()
sql = "update item_cart set quantity= %s where product_id = %s;" % (3, 382)
cursor.execute(sql)
cursor.fetchall()
```

```
()
```

```python
# 장바구니 열람
cursor = hm_db.cursor()

sql = "select * from item_cart"
cursor.execute(sql)
cursor.fetchall()
```

```
((200, 1, 45924, 1),
 (220, 3, 88392, 1),
 (230, 2, 372, 11),
 (240, 7, 74022, 111),
 (255, 3, 382, 1))
```

```
# order_items
sql = "insert into order_items values(%s,%s);" % (7676,382)
cursor.execute(sql)
hm_db.commit()
cursor.fetchall()

# order_details 작성
sql = "insert into order_details values(%s,%s,%s,%s,%s)" % (8787, 382, 49000*3, 1, 7676)
cursor.execute(sql)
hm_db.commit()
cursor.fetchall()
```

```
()
```

```
# order detail 열람
cursor = hm_db.cursor()

sql = "select * from order_details"
cursor.execute(sql)
cursor.fetchall()
```

```
((8787, 382, 147000, 1, 7676),)
```

```
# payment_details 작성
sql = "Insert into payment_details values(3535,3,'HM','paid','KR00',8787)"
cursor.execute(sql)
hm_db.commit()
cursor.fetchall()

# payment_details 열람
sql = "select * from payment_details"
cursor.execute(sql)
hm_db.commit()
cursor.fetchall()
```

```
((3535, 3, 'HM', 'paid', 'KR00', 8787),)
```

3. show that your program is working correctly. In case you don't use GUI(graphic user interface), make sure you provide a proof that your program is working correctly.
   For example, suppose the task is about showing a summary report for a given input.
   Then you should clearly demonstrate that your program correctly displays all necessary information when a certain input is given. There will be extra points for GUI development, though.

O You should upload the following files at e-class by due date:

(1) database dump file

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/f6be9b1d-8550-46d3-9e25-64bda81bef05/hmdb_dump.sql

```
-- MySQL dump 10.13  Distrib 8.0.27, for macos11 (x86_64)
--
```

```
-- Host: localhost    Database: hmdb
-- ------------------------------------------------------
-- Server version 8.0.27

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!50503 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

--
-- Table structure for table `discount`
--

DROP TABLE IF EXISTS `discount`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `discount` (
  `discount_id` int NOT NULL,
  `discount_percent` int DEFAULT NULL,
  `discount_desc` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`discount_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `discount`
--

LOCK TABLES `discount` WRITE;
/*!40000 ALTER TABLE `discount` DISABLE KEYS */;
INSERT INTO `discount` VALUES (1,13,'shoes'),(242,22,'bags'),(28489,4,'shorts'),(33902,30,'wallets'),(45354,63,'hoodies');
/*!40000 ALTER TABLE `discount` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `item_cart`
--

DROP TABLE IF EXISTS `item_cart`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `item_cart` (
  `item_cart_id` int NOT NULL,
  `quantity` int DEFAULT NULL,
  `product_id` int DEFAULT NULL,
  `user_id` int DEFAULT NULL,
  PRIMARY KEY (`item_cart_id`),
  KEY `product_id` (`product_id`),
  KEY `user_id` (`user_id`),
  CONSTRAINT `item_cart_ibfk_1` FOREIGN KEY (`product_id`) REFERENCES `product` (`product_id`),
  CONSTRAINT `item_cart_ibfk_2` FOREIGN KEY (`user_id`) REFERENCES `user` (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `item_cart`
--

LOCK TABLES `item_cart` WRITE;
/*!40000 ALTER TABLE `item_cart` DISABLE KEYS */;
INSERT INTO `item_cart` VALUES (200,1,45924,1),(220,3,88392,3),(230,2,372,4),(240,7,74022,5),(255,3,382,1);
/*!40000 ALTER TABLE `item_cart` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `order_details`
--

DROP TABLE IF EXISTS `order_details`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `order_details` (
  `order_details_id` int NOT NULL,
  `product_number` int DEFAULT NULL,
  `total_price` int DEFAULT NULL,
  `user_id` int DEFAULT NULL,
  `order_items_id` int DEFAULT NULL,
  PRIMARY KEY (`order_details_id`),
  KEY `user_id` (`user_id`),
  KEY `order_items_id` (`order_items_id`),
  CONSTRAINT `order_details_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `user` (`user_id`),
  CONSTRAINT `order_details_ibfk_2` FOREIGN KEY (`order_items_id`) REFERENCES `order_items` (`order_items_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `order_details`
--

LOCK TABLES `order_details` WRITE;
/*!40000 ALTER TABLE `order_details` DISABLE KEYS */;
INSERT INTO `order_details` VALUES (8787,382,147000,1,7676);
/*!40000 ALTER TABLE `order_details` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `order_items`
--

DROP TABLE IF EXISTS `order_items`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `order_items` (
  `order_items_id` int NOT NULL,
  `product_id` int DEFAULT NULL,
  PRIMARY KEY (`order_items_id`),
  KEY `product_id` (`product_id`),
  CONSTRAINT `order_items_ibfk_1` FOREIGN KEY (`product_id`) REFERENCES `product` (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `order_items`
--

LOCK TABLES `order_items` WRITE;
/*!40000 ALTER TABLE `order_items` DISABLE KEYS */;
INSERT INTO `order_items` VALUES (7676,382);
/*!40000 ALTER TABLE `order_items` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `payment_details`
--
```

```
DROP TABLE IF EXISTS `payment_details`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `payment_details` (
  `payment_id` int NOT NULL,
  `amount` int DEFAULT NULL,
  `provider` varchar(20) DEFAULT NULL,
  `status` varchar(20) DEFAULT NULL,
  `receipt` varchar(20) DEFAULT NULL,
  `order_details_id` int DEFAULT NULL,
  PRIMARY KEY (`payment_id`),
  KEY `order_details_id` (`order_details_id`),
  CONSTRAINT `payment_details_ibfk_1` FOREIGN KEY (`order_details_id`) REFERENCES `order_details` (`order_details_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `payment_details`
--

LOCK TABLES `payment_details` WRITE;
/*!40000 ALTER TABLE `payment_details` DISABLE KEYS */;
INSERT INTO `payment_details` VALUES (3535,3,'HM','paid','KR00',8787);
/*!40000 ALTER TABLE `payment_details` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `product`
--

DROP TABLE IF EXISTS `product`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `product` (
  `product_id` int NOT NULL,
  `product_name` varchar(20) DEFAULT NULL,
  `price` int DEFAULT NULL,
  `category` varchar(20) DEFAULT NULL,
  `product_desc` varchar(20) DEFAULT NULL,
  `sku` varchar(20) DEFAULT NULL,
  PRIMARY KEY (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `product`
--

LOCK TABLES `product` WRITE;
/*!40000 ALTER TABLE `product` DISABLE KEYS */;
INSERT INTO `product` VALUES (372,'black wallet',100000,'wallets','for business men','W384248'),(382,'blue_jacket',49000,'jackets','for men','JK102476'),(34892,'students bag',40000,'bags','for students','B
/*!40000 ALTER TABLE `product` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Table structure for table `user`
--

DROP TABLE IF EXISTS `user`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!50503 SET character_set_client = utf8mb4 */;
CREATE TABLE `user` (
  `user_id` int NOT NULL,
  `username` varchar(20) DEFAULT NULL,
  `name` varchar(20) DEFAULT NULL,
  `gender` varchar(5) DEFAULT NULL,
  `address` varchar(30) DEFAULT NULL,
  `telephone_number` varchar(13) DEFAULT NULL,
  PRIMARY KEY (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `user`
--

LOCK TABLES `user` WRITE;
/*!40000 ALTER TABLE `user` DISABLE KEYS */;
INSERT INTO `user` VALUES (1,'hhshin','신현호','M','경기도 수원시 영통구','01000000000'),(2,'gunnoo','이건우','M','서울특별시 마포구','010000000001'),(3,'Kindess','박친절','F','대구광역시 달성구','01000000011'),(4,'James98'
/*!40000 ALTER TABLE `user` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2021-11-17 21:39:52
```

(2) program file

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/e717f9ea-7c5d-41c7-9178-6230afbd7f4f/hmdb.sql

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/ee9cf693-812a-4c7c-b92e-b3ff00760ec7/hmdb_python.py

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/459a91af-30de-4860-a947-2d5f4b367d5b/hmdb_python.ipynb

(3) report file

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/477bf12d-037f-4e6e-9d9e-8126a9f1a4af/Database_Systems_Laboratory_Homework_3.pdf

(4) output file, etc (every file necessary to test your program)

O Reminder: Your homework grade depends on your contribution to the project. No contribution, no grade.
Due : Nov 24 (Wed)