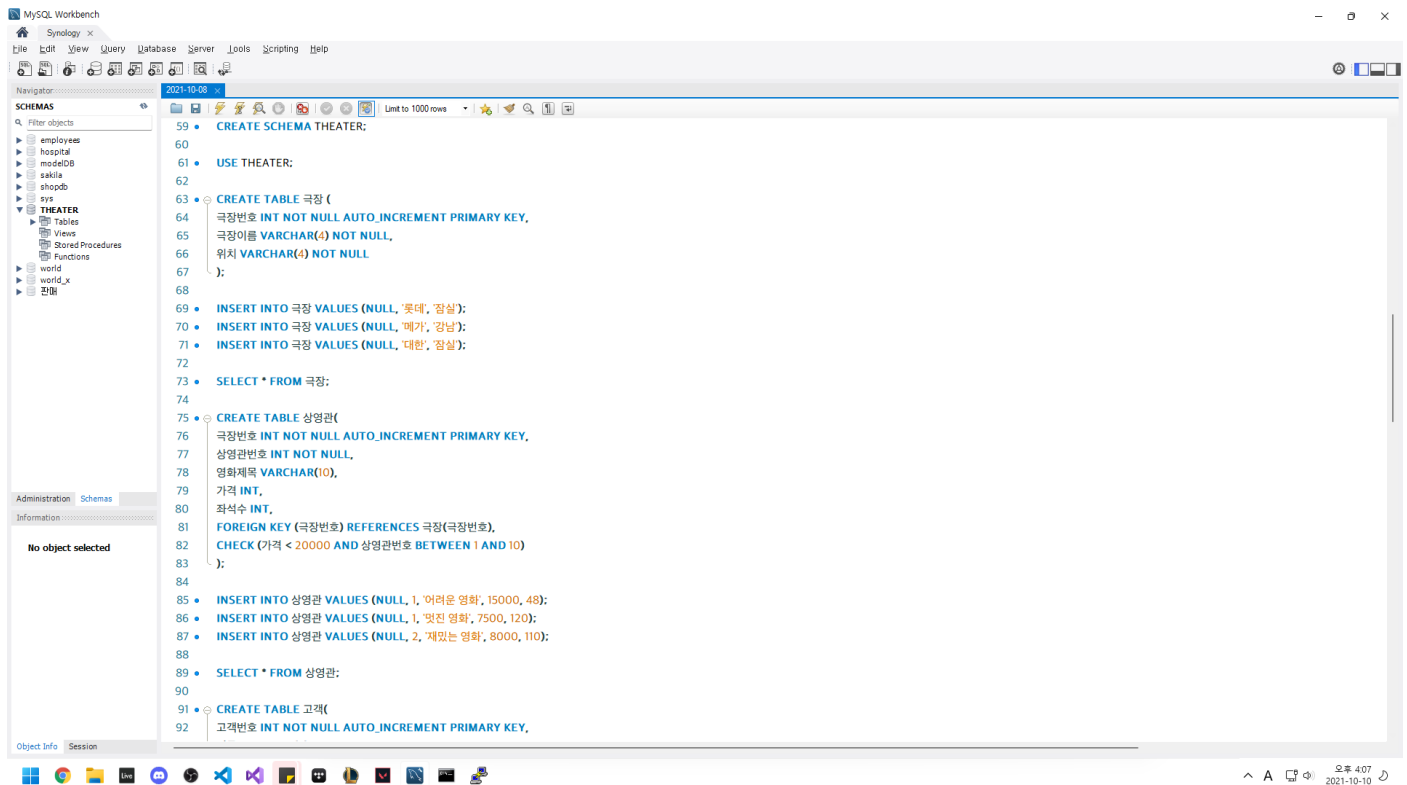


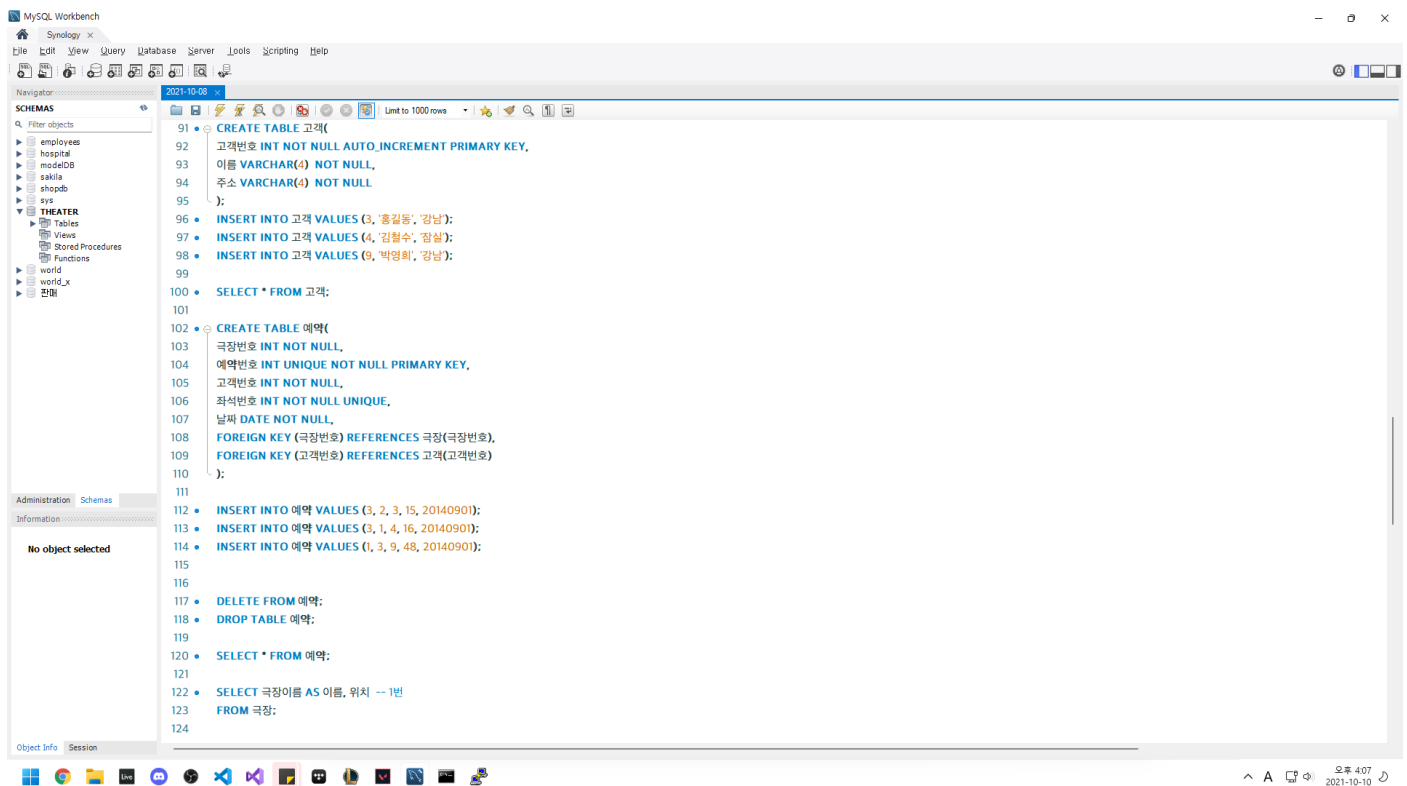
## <7장 DDL+DML 실습> -- 201720970\_권대한

### <스키마 생성, 선언, 테이블 생성, 데이터 삽입>



MySQL Workbench interface showing SQL queries for creating and inserting data into the 'THEATER' schema.

```
59 • CREATE SCHEMA THEATER;
60
61 • USE THEATER;
62
63 • CREATE TABLE 극장 (
64   극장번호 INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
65   극장이름 VARCHAR(4) NOT NULL,
66   위치 VARCHAR(4) NOT NULL
67 );
68
69 • INSERT INTO 극장 VALUES (NULL, '롯데', '잠실');
70 • INSERT INTO 극장 VALUES (NULL, '메가', '강남');
71 • INSERT INTO 극장 VALUES (NULL, '대한', '잠실');
72
73 • SELECT * FROM 극장;
74
75 • CREATE TABLE 상영관(
76   극장번호 INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
77   상영관번호 INT NOT NULL,
78   영화제목 VARCHAR(10),
79   가격 INT,
80   좌석수 INT,
81   FOREIGN KEY (극장번호) REFERENCES 극장(극장번호),
82   CHECK (가격 < 20000 AND 상영관번호 BETWEEN 1 AND 10)
83 );
84
85 • INSERT INTO 상영관 VALUES (NULL, 1, '어려운 영화', 15000, 48);
86 • INSERT INTO 상영관 VALUES (NULL, 1, '멋진 영화', 7500, 120);
87 • INSERT INTO 상영관 VALUES (NULL, 2, '재밌는 영화', 8000, 110);
88
89 • SELECT * FROM 상영관;
90
91 • CREATE TABLE 고객(
92   고객번호 INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
```



MySQL Workbench interface showing SQL queries for creating and inserting data into the 'THEATER' schema.

```
91 • CREATE TABLE 고객(
92   고객번호 INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
93   이름 VARCHAR(4) NOT NULL,
94   주소 VARCHAR(4) NOT NULL
95 );
96 • INSERT INTO 고객 VALUES (3, '홍길동', '강남');
97 • INSERT INTO 고객 VALUES (4, '김철수', '잠실');
98 • INSERT INTO 고객 VALUES (9, '박영희', '강남');
99
100 • SELECT * FROM 고객;
101
102 • CREATE TABLE 예약(
103   극장번호 INT NOT NULL,
104   예약번호 INT UNIQUE NOT NULL PRIMARY KEY,
105   고객번호 INT NOT NULL,
106   좌석번호 INT NOT NULL UNIQUE,
107   날짜 DATE NOT NULL,
108   FOREIGN KEY (극장번호) REFERENCES 극장(극장번호),
109   FOREIGN KEY (고객번호) REFERENCES 고객(고객번호)
110 );
111
112 • INSERT INTO 예약 VALUES (3, 2, 3, 15, 20140901);
113 • INSERT INTO 예약 VALUES (3, 1, 4, 16, 20140901);
114 • INSERT INTO 예약 VALUES (1, 3, 9, 48, 20140901);
115
116
117 • DELETE FROM 예약;
118 • DROP TABLE 예약;
119
120 • SELECT * FROM 예약;
121
122 • SELECT 극장이름 AS 이름, 위치 -- 1번
123   FROM 극장;
124
```

## <단순 질의 1-5 출력화면>

MySQL Workbench interface showing SQL queries and their results.

**Navigation:** Schemas, Filter objects, employees, hospital, modelDB, sakila, shopdb, sys, THEATER, Tables, Views, Stored Procedures, Functions, world, world\_x, 판매

**SQL Editor:**

```

122 • SELECT 극장이름 AS 이름, 위치 -- 1번
123 FROM 극장;
124
125 • SELECT *
126 FROM 극장
127 WHERE 극장.위치 = '잠실';
128
129 • SELECT *
130 FROM 고객
131 WHERE 고객.주소 = '잠실'
132 ORDER BY 이름 ASC;
133
134 • SELECT 극장번호, 상영관번호, 영화제목
135 FROM 상영관
136 WHERE 가격 <= 8000;
137
138 • SELECT *
139 FROM 고객
140 WHERE 고객.주소 IN (SELECT 위치
141                      FROM 극장)
142 ORDER BY 주소 ASC;
143
144 • SELECT COUNT(*) AS '극장의 수' -- 2번
145 FROM 극장;
146
147 • SELECT AVG(가격) AS '상영되는 영화의 평균 가격'
148 FROM 상영관;
  
```

**Result Grid:**

고객 번호	이름	주소
3	홍길동	강남
9	박영희	강남
4	김철수	잠실

Object Info: Session, 고객 13 x

## <집계 질의 2-2 출력화면>

MySQL Workbench interface showing SQL queries and their results.

**Navigation:** Schemas, Filter objects, employees, hospital, modelDB, sakila, shopdb, sys, THEATER, Tables, Views, Stored Procedures, Functions, world, world\_x, 판매

**SQL Editor:**

```

144 • SELECT COUNT(*) AS '극장의 수' -- 2번
145 FROM 극장;
146
147 • SELECT AVG(가격) AS '상영되는 영화의 평균 가격'
148 FROM 상영관;
149
150 • SELECT 영화제목 -- 3번
151 FROM 상영관
152 WHERE 극장번호 IN (SELECT 극장번호
153                   FROM 극장
154                   WHERE 극장이름 = '대한');
155
156 • SELECT 이름
157 FROM 고객 JOIN 예약 ON 고객.고객번호 = 예약.고객번호
158 WHERE 극장번호 IN (SELECT 극장번호
159                   FROM 극장
160                   WHERE 극장이름 = '대한');
161
162 • SELECT SUM(가격) AS '대한 극장의 전체 수입'
163 FROM 예약 JOIN 상영관 ON 예약.극장번호 = 상영관.극장번호
164 WHERE 상영관.극장번호 IN (SELECT 극장번호
165                           FROM 극장
166                           WHERE 극장.극장이름 = '대한');
167
168 • SELECT SUM(가격) AS '대한 극장의 전체 수입' -- ambiguous error
169 FROM 예약 JOIN 상영관 ON 예약.극장번호 = 상영관.극장번호
170 WHERE 예약.극장번호 IN (SELECT 극장번호
  
```

**Result Grid:**

상영되는 영화의 평균 가격
10166.6667

Object Info: Session, Result 14 x

## <부속 질의와 조인 3-3 출력화면>

MySQL Workbench interface showing a SQL query and its result. The query is as follows:

```

150 • SELECT 영화제목 -- 3번
151 FROM 상영관
152 WHERE 극장번호 IN (SELECT 극장번호
153 FROM 극장
154 WHERE 극장이름 = '대한');
155
156 • SELECT 이름
157 FROM 예약 JOIN 예약 ON 고객,고객번호 = 예약,고객번호
158 WHERE 극장번호 IN (SELECT 극장번호
159 FROM 극장
160 WHERE 극장이름 = '대한');
161
162 • SELECT SUM(가격) AS '대한 극장의 전체 수입'
163 FROM 예약 JOIN 상영관 ON 예약,극장번호 = 상영관,극장번호
164 WHERE 상영관,극장번호 IN (SELECT 극장번호
165 FROM 극장
166 WHERE 극장,극장이름 = '대한');
167
168 • SELECT SUM(가격) AS '대한 극장의 전체 수입' -- ambiguous error
169 FROM 예약 JOIN 상영관 ON 예약,극장번호 = 상영관,극장번호
170 WHERE 예약,극장번호 IN (SELECT 극장번호
171 FROM 극장
172 WHERE 극장,극장이름 = '대한');
173
174 • SELECT 극장이름, COUNT(상영관번호) AS '극장별 상영관의 수' -- 4번
175 FROM 상영관 JOIN 극장
176 GROUP BY 극장이름;

```

The result grid shows the following data:

대한 극장의 전체 수입
16000

## <그룹질의 4-3 출력화면>

MySQL Workbench interface showing a SQL query and its result. The query is as follows:

```

165 FROM 극장
166 WHERE 극장,극장이름 = '대한';
167
168 • SELECT SUM(가격) AS '대한 극장의 전체 수입' -- ambiguous error
169 FROM 예약 JOIN 상영관 ON 예약,극장번호 = 상영관,극장번호
170 WHERE 예약,극장번호 IN (SELECT 극장번호
171 FROM 극장
172 WHERE 극장,극장이름 = '대한');
173
174 • SELECT 극장이름, COUNT(상영관번호) AS '극장별 상영관의 수' -- 4번
175 FROM 상영관 JOIN 극장
176 GROUP BY 극장이름;
177
178 • SELECT *
179 FROM 극장 JOIN 상영관 ON 극장,극장번호 = 상영관,극장번호
180 WHERE 극장,위치= '잠실';
181
182 • SELECT 극장이름, COUNT(극장이름) AS '극장별 관람 고객의 수'
183 FROM 예약 JOIN 극장 ON 예약,극장번호 = 극장,극장번호
184 WHERE 예약,날짜= '2014-09-01'
185 GROUP BY 극장이름;
186
187
188 • UPDATE 상영관 SET 가격 = 가격 * 1.1; -- 5번
189 • SELECT * FROM 상영관
190 ORDER BY 영화제목;

```

The result grid shows the following data:

극장이름	극장별 관람 고객의 수
대한	2
롯데	1

## <DML 5 출력화면>

The screenshot displays the MySQL Workbench interface. The main editor window contains a SQL script with the following queries:

```
165 FROM 극장
166 WHERE 극장.극장이름 = '대한';
167
168 • SELECT SUM(가격) AS '대한 극장의 전체 수입' -- ambiguous error
169 FROM 예약 JOIN 상영관 ON 예약.극장번호 = 상영관.극장번호
170 WHERE 예약.극장번호 IN (SELECT 극장번호
171 FROM 극장
172 WHERE 극장.극장이름 = '대한');
173
174 • SELECT 극장이름, COUNT(상영관번호) AS '극장별 상영관의 수' -- 4번
175 FROM 상영관 JOIN 극장
176 GROUP BY 극장이름;
177
178 • SELECT *
179 FROM 극장 JOIN 상영관 ON 극장.극장번호 = 상영관.극장번호
180 WHERE 극장.위치 = '잠실';
181
182 • SELECT 극장이름, COUNT(극장이름) AS '극장별 관람 고객의 수'
183 FROM 예약 JOIN 극장 ON 예약.극장번호 = 극장.극장번호
184 WHERE 예약.날짜 = '2014-09-01'
185 GROUP BY 극장이름;
186
187
188 • UPDATE 상영관 SET 가격 = 가격 * 1.1; -- 5번
189 • SELECT * FROM 상영관
190 ORDER BY 영화제목;
```

Below the script, the 'Result Grid' shows the output of the last query (SELECT \* FROM 상영관 ORDER BY 영화제목):

극장번호	상영관번호	영화제목	가격	좌석수
2	1	멋진 영화	7500	120
1	1	어려운 영화	15000	48
3	2	재밌는 영화	8000	110

The interface also shows the 'SCHEMAS' panel on the left, the 'Administration' and 'Schemas' tabs, and the 'Object Info' and 'Session' panels at the bottom.