

데이터관리와 분석

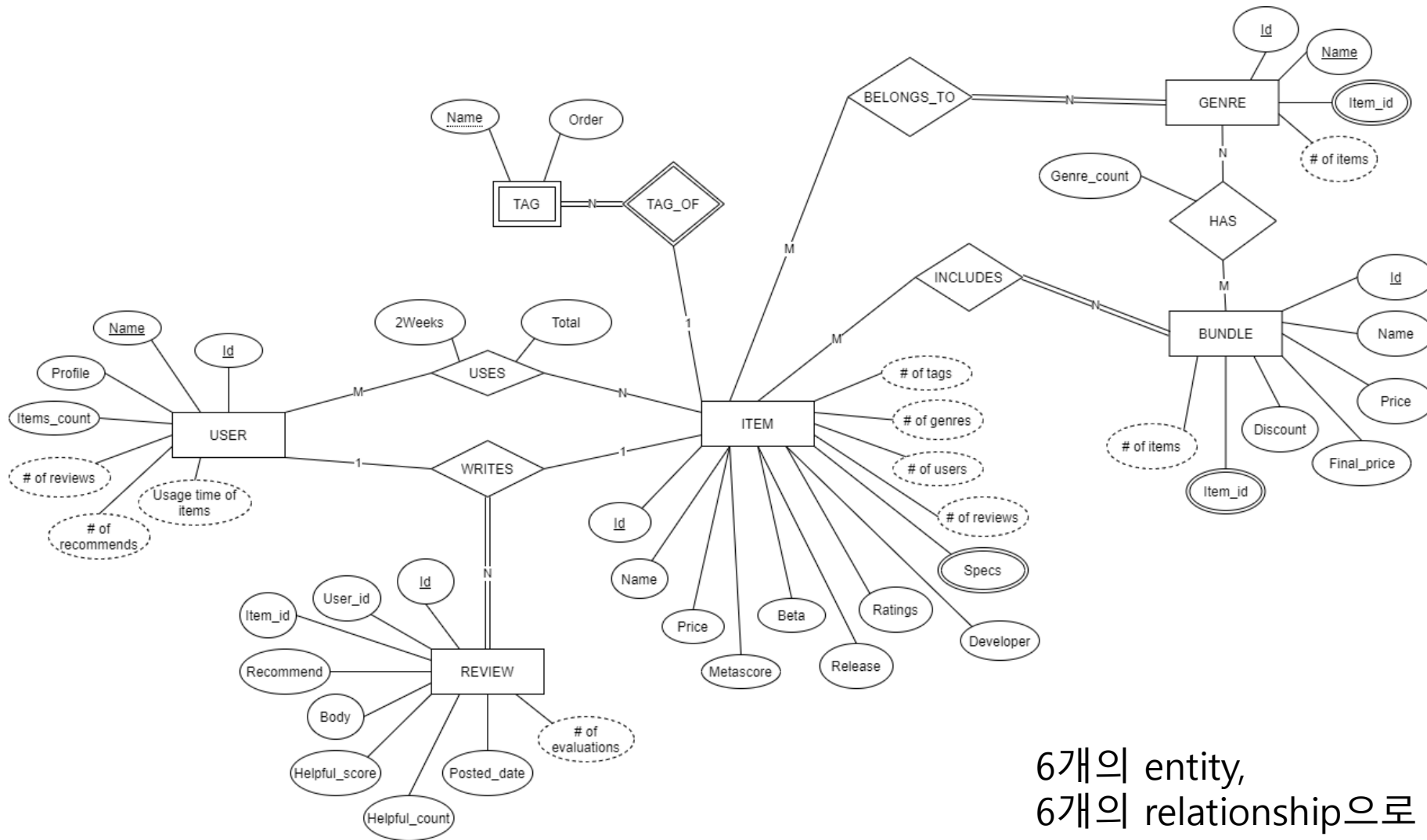


# Project #1

1조 김세찬, 이건희, 홍지현, 추성민



## Part 1. ER diagram 도식화

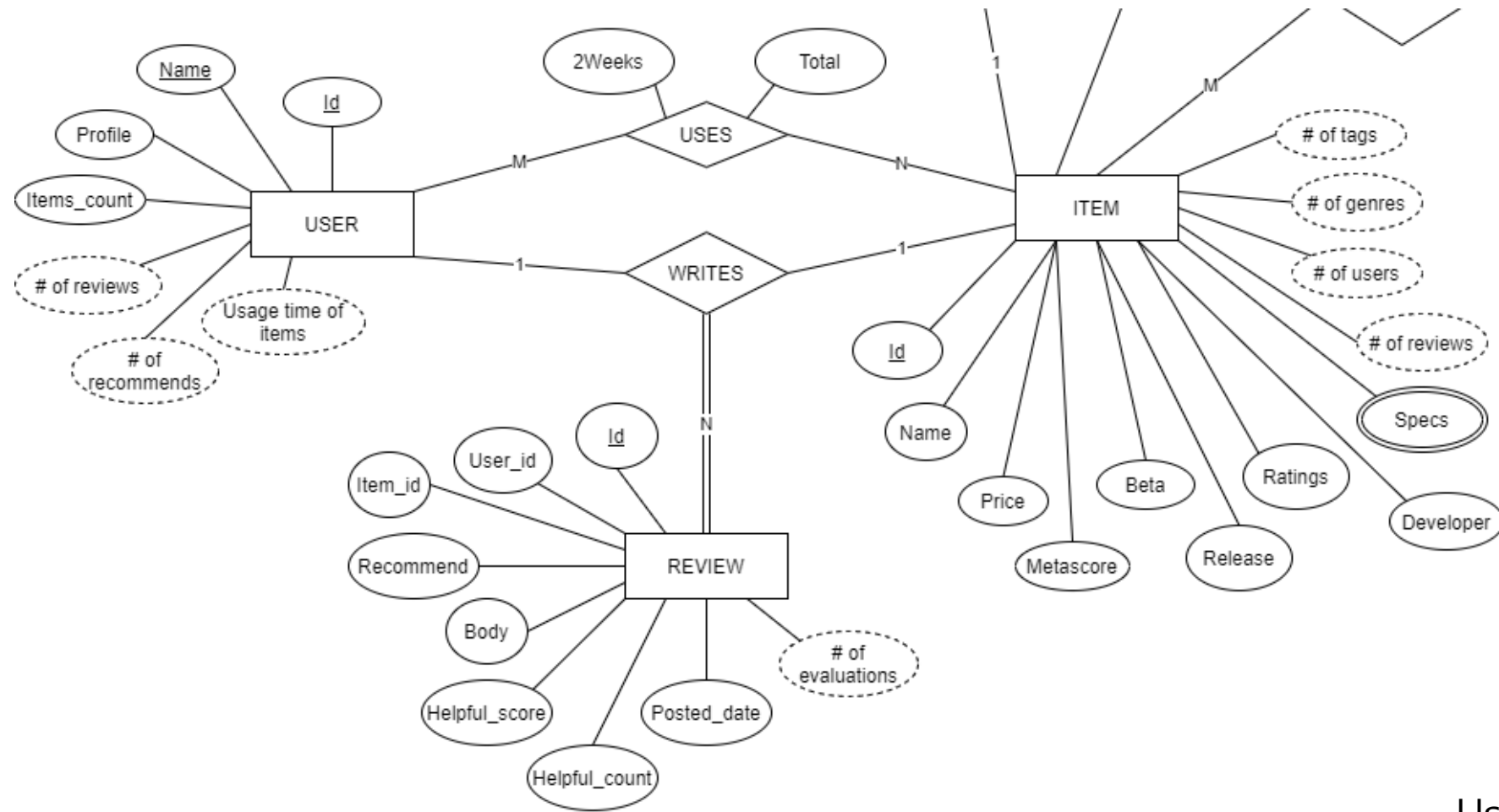


6개의 entity,  
6개의 relationship으로 DB설계



## Part 1. ER diagram 도식화

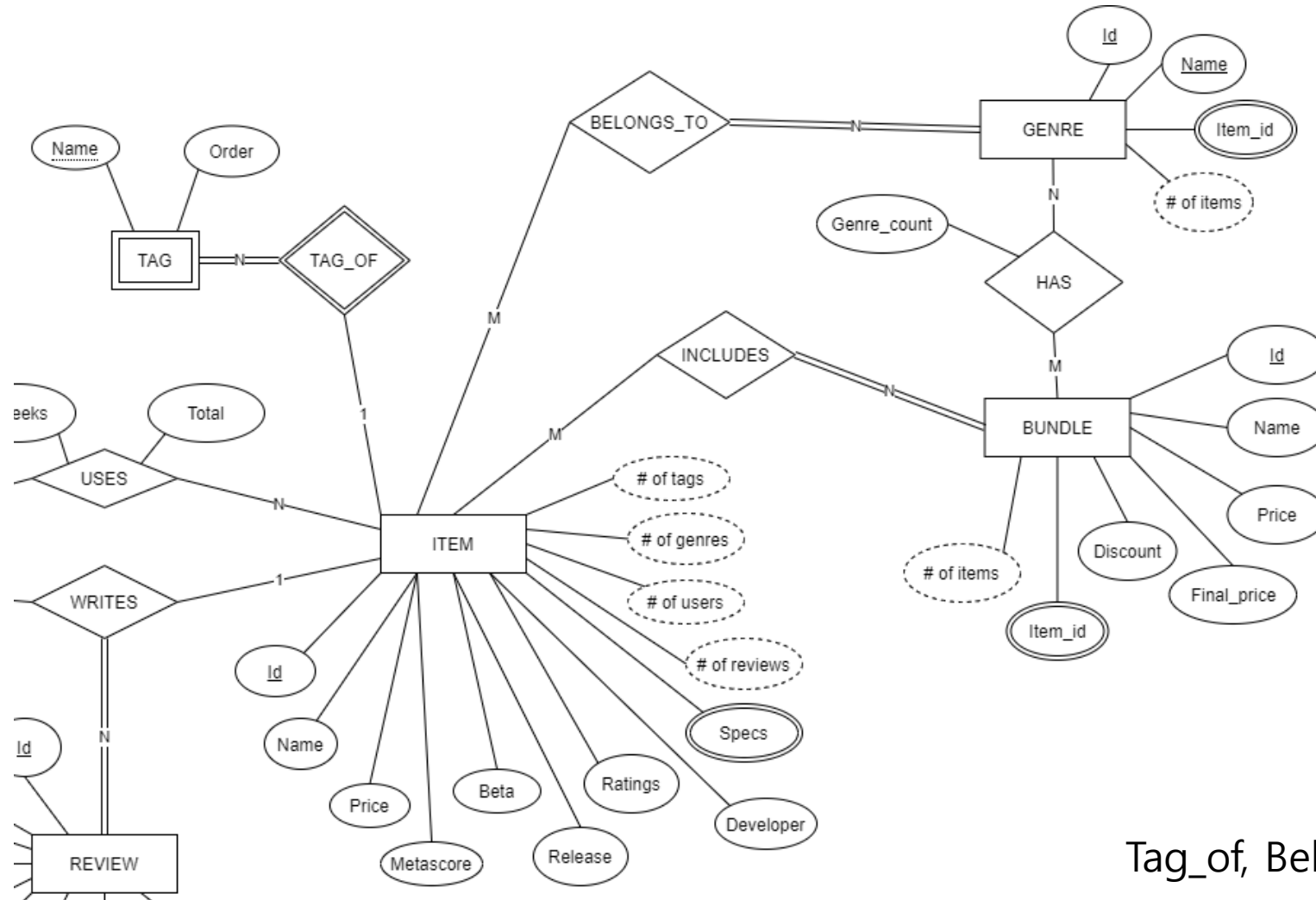
• • •



Uses, Writes relationship

## Part 1. ER diagram 도식화

• • •



Tag\_of, Belongs\_To, Includes, Has relationship

## Part 2. DB 구현 및 데이터 입력



```
31 # 1. user table
32 cursor.execute('''
33 CREATE TABLE IF NOT EXISTS USER(
34 id BIGINT(20) NOT NULL,
35 user_name VARCHAR(255) NOT NULL,
36 profile TINYINT(1) NOT NULL,
37 items_count INT(11) NOT NULL,
38 PRIMARY KEY(id),
39 UNIQUE(user_name));
40 ''')
41
42 # 2. item table
43 cursor.execute('''
44 CREATE TABLE IF NOT EXISTS ITEM(
45 id BIGINT(20) NOT NULL,
46 item_name VARCHAR(255) NOT NULL,
47 price FLOAT,
48 beta_version TINYINT(1) NOT NULL,
49 ratings INT(11) NOT NULL,
50 metascore INT(11) NOT NULL,
51 developers VARCHAR(255),
52 release_date DATE,
53 PRIMARY KEY(id));
54 ''')
55
56 # 3. user_item table
57 cursor.execute('''
58 CREATE TABLE IF NOT EXISTS USER_ITEM(
59 user_id BIGINT(20) NOT NULL,
60 item_id BIGINT(20) NOT NULL,
61 usagetime_2weeks INT(11) NOT NULL,
62 usagetime_total INT(11) NOT NULL);
63 ''')
64
```

Tag를 제외한 5개의 Entity의 경우  
각자의 ID를 primary key로 설정,  
Tag는 weak entity type이므로  
(item\_id, tag\_name)을 primary key로 설정

item table의  
price, developers, release\_date,  
그리고 item\_spec table의 spec\_name을 제외한  
모든 column에 not null 조건을 부여

## Part 2. DB 구현 및 데이터 입력

```
141 # Requirement3: insert data
142 def requirement3(host, user, password, directory):
143     cnx = mysql.connector.connect(host=host, user=user, password=password)
144     cursor = cnx.cursor()
145     cursor.execute('SET GLOBAL innodb_buffer_pool_size=2*1024*1024*1024;')
146
147     # TODO: WRITE CODE HERE
148     cursor.execute('CREATE DATABASE IF NOT EXISTS DMA_team01;')
149     cursor.execute('USE DMA_team01;')
150
151     csv_dict = {'bundle_genre.csv': ['INT', 'VARCHAR', 'INT'],
152                'bundle_item.csv': ['INT', 'INT'],
153                'bundle.csv': ['INT', 'VARCHAR', 'FLOAT', 'FLOAT', 'FLOAT'],
154                'genre.csv': ['VARCHAR', 'VARCHAR'],
155                'item_genre.csv': ['INT', 'VARCHAR'],
156                'item_specs.csv': ['INT', 'VARCHAR'],
157                'item.csv': ['BIGINT', 'VARCHAR', 'FLOAT', 'TINYINT', 'INT', 'INT', 'VARCHAR', 'DATE'],
158                'review.csv': ['VARCHAR', 'BIGINT', 'INT', 'TINYINT', 'INT', 'FLOAT', 'INT', 'DATE'],
159                'tag.csv': ['INT', 'VARCHAR', 'INT'],
160                'user_item.csv': ['BIGINT', 'INT', 'INT', 'INT'],
161                'user.csv': ['BIGINT', 'VARCHAR', 'TINYINT', 'INT']}
```

자료형을 dictionary 형태로 저장하여  
각 column별 데이터 타입을 개별적으로 처리함

## Part 2. DB 구현 및 데이터 입력

```
179 for idx, r in enumerate(row):
180     if len(r) == 0: # r이 빈 string일 경우 'null' 입력
181         row[idx] = 'null'
182         continue
183     if key == 'item.csv' and r == 'nan': # item table에서 r이 nan일 경우 'null'입력
184         row[idx] = 'null'
185         continue
186     if type_list[idx] in ['TINYINT', 'INT', 'BIGINT']: # r이 정수면 int형으로
187         row[idx] = int(r)
188     elif type_list[idx] in ['FLOAT']: # r이 실수면 float형으로
189         try:
190             row[idx] = float(r)
191         except:
192             if r[-1] == '%': # 퍼센트 자료형의 경우 % 제거 후 실수로 변경
193                 row[idx] = float(r[:-1])
194             elif r[0] == '$': # 가격의 경우 $제거 후 실수로 변경
195                 row[idx] = float(r[1:])
196     elif type_list[idx] in ['VARCHAR', 'DATE']: # r이 문자열이면 string형으로
197         row[idx] = r
198     else:
199         raise ValueError
```

## Part 2. DB 구현 및 데이터 입력

```
212 # Requirement4: add constraint (foreign key)
213 def requirement4(host, user, password):
214     cnx = mysql.connector.connect(host=host, user=user, password=password)
215     cursor = cnx.cursor()
216     cursor.execute('SET GLOBAL innodb_buffer_pool_size=2*1024*1024*1024;')
217
218     # TODO: WRITE CODE HERE
219     cursor.execute('CREATE DATABASE IF NOT EXISTS DMA_team01;')
220     cursor.execute('USE DMA_team01;')
221
222     # review의 FOREIGN KEY 설정
223     cursor.execute('ALTER TABLE REVIEW ADD CONSTRAINT FOREIGN KEY(user_id) REFERENCES USER(id);')
224     cursor.execute('ALTER TABLE REVIEW ADD CONSTRAINT FOREIGN KEY(item_id) REFERENCES ITEM(id);')
225
226     # user_item의 FOREIGN KEY 설정
227     cursor.execute('ALTER TABLE USER_ITEM ADD CONSTRAINT FOREIGN KEY(user_id) REFERENCES USER(id);')
228     cursor.execute('ALTER TABLE USER_ITEM ADD CONSTRAINT FOREIGN KEY(item_id) REFERENCES ITEM(id);')
229
230     # item_genre의 FOREIGN KEY 설정
231     cursor.execute('ALTER TABLE ITEM_GENRE ADD CONSTRAINT FOREIGN KEY(item_id) REFERENCES ITEM(id);')
232     cursor.execute('ALTER TABLE ITEM_GENRE ADD CONSTRAINT FOREIGN KEY(genre_id) REFERENCES GENRE(id);')
233
234     # tag의 FOREIGN KEY 설정
235     cursor.execute('ALTER TABLE TAG ADD CONSTRAINT FOREIGN KEY(item_id) REFERENCES ITEM(id);')
236
237     # genre의 FOREIGN KEY 설정
238     cursor.execute('ALTER TABLE BUNDLE_GENRE ADD CONSTRAINT FOREIGN KEY(genre_id) REFERENCES GENRE(id);')
239
240     # bundle의 FOREIGN KEY 설정
241     cursor.execute('ALTER TABLE BUNDLE_ITEM ADD CONSTRAINT FOREIGN KEY(bundle_id) REFERENCES BUNDLE(id);')
```



## Part 2. DB 구현 및 데이터 입력

• • •

### USER

<u>id</u>	user_name	profile_image	items_count
-----------	-----------	---------------	-------------

### ITEM

<u>id</u>	item_name	price	beta_version	ratings	metascore	developer	release_data
-----------	-----------	-------	--------------	---------	-----------	-----------	--------------

### USER\_ITEM

<u>user_id</u>	<u>item_id</u>	usagetime_2weeks	usagetime_total
----------------	----------------	------------------	-----------------

### REVIEW

<u>id</u>	<u>user_id</u>	<u>item_id</u>	recommend	body	helpful_score	helpful_count	posted_date
-----------	----------------	----------------	-----------	------	---------------	---------------	-------------

### GENRE

<u>id</u>	genre_name
-----------	------------

### ITEM\_GENRE

<u>item_id</u>	<u>genre_id</u>
----------------	-----------------

### BUNDLE

<u>id</u>	bundle_name	price	final_price	discount
-----------	-------------	-------	-------------	----------

### BUNDLE\_ITEM

<u>bundle_id</u>	<u>item_id</u>
------------------	----------------

### BUNDLE\_GENRE

<u>bundle_id</u>	<u>genre_id</u>	genre_count
------------------	-----------------	-------------

### TAG

<u>item_id</u>	<u>tag_name</u>	tag_order
----------------	-----------------	-----------

### ITEM\_SPECS

<u>item_id</u>	<u>spec_name</u>
----------------	------------------

...

**Thank you :-)**