

課題5

1029332978 工学部情報学科計算機科学コース 3回 上野山遼音

課題4で更新したDBに対して,

1. 関係代数の射影および選択に対応するSQL文
2. 関係代数の自然結合に対応するSQL文
3. UNIONを含むSQL文
4. EXCEPTを含むSQL文
5. DISTINCTを含むSQL文
6. 集合関数(COUNT,SUM,AVG,MAX,MIN)を用いたSQL文
7. 副質問(sub query)を含むSQL文
8. UPDATEを含むSQL文
9. ORDER BYを含むSQL文
10. CREATE VIEWを含むSQL文

の内容のsql文を作成して実行する.

1 射影及び選択に対応するsql文

```
select player_name from players;
```

説明

プレイヤーの名簿を出力する.

実行結果

```
はるし  
港悪亜
```

2 関係代数の自然結合に対応するSQL文

```
select * from songs natural inner join charts
```

説明

songsスキーマとchartsスキーマを内結合させる.

実行結果

```
1|ボリエール社畜|苦茶の音楽工房|path/to/mp3|1|10|譜面-2147483648号
2|君が代|林廣守|path/to/wav|2|9|譜面-2147483648号
1|ボリエール社畜|苦茶の音楽工房|path/to/mp3|3|7|はるし
1|ボリエール社畜|苦茶の音楽工房|path/to/mp3|4|0|はるし
```

3 UNIONを含むSQL文

```
SELECT distinct songs.title, charts.chart_designer
FROM songs
INNER JOIN charts ON songs.song_id = charts.song_id
WHERE charts.chart_id IN (SELECT chart_id FROM playData WHERE favorites >=
1)

UNION

select distinct songs.title, charts.chart_designer
FROM songs
INNER JOIN charts ON songs.song_id = charts.song_id
WHERE charts.chart_id IN (SELECT chart_id FROM playData WHERE best_score =
1000000);
```

説明

理論値を取得されている楽曲と、お気に入りに登録されている楽曲の和集合を求める。(人気ではありそう)

実行結果

```
ボリエール無職|譜面-2147483648号
```

4 EXCEPTを含むSQL文

```
select songs.title, charts.chart_designer from songs, charts
except
SELECT distinct songs.title, charts.chart_designer
FROM songs
INNER JOIN charts ON songs.song_id = charts.song_id
WHERE charts.chart_id IN (SELECT chart_id FROM playData WHERE favorites =
1);
```

説明

誰もお気に入りに登録していない曲のタイトルと譜面制作者を出力する。

実行結果

```
ボリエール社畜 | はるし  
君が代 | はるし  
君が代 | 譜面-2147483648号
```

5 DISTINCTを含むSQL文

```
select Distinct title, chart_designer from songs natural inner join charts
```

説明

楽曲と譜面制作者の対応のリストを出力する.

実行結果

```
ボリエール社畜 | 譜面-2147483648号  
君が代 | 譜面-2147483648号  
ボリエール社畜 | はるし
```

6 集合関数(COUNT,SUM,AVG,MAX,MIN)を用いたSQL文

```
select count(distinct player_id) from playData where best_score=1000000;
```

説明

何かしらの曲で1000000点(理論値)を達成したことのある人数を求めるsql文.

実行結果

```
1(すくな!)
```

7 副質問(sub query)を含むSQL文

```
SELECT distinct songs.title, charts.chart_designer  
FROM songs  
INNER JOIN charts ON songs.song_id = charts.song_id  
WHERE charts.chart_id IN (SELECT chart_id FROM playData WHERE favorites =  
1);
```

説明

誰かのお気に入りに登録されている楽曲の詳細情報を求める.

実行結果

```
ボリエール社畜 | 譜面-2147483648号
```

8 UPDATEを含むSQL文

```
UPDATE songs SET title = 'トロピカル無職' WHERE title = 'ボリエール社畜';
```

説明

誤って入力されていた楽曲名を修正するためのsql文.

実行結果

```
(出力なし)
```

select * from songsを実行すると、実際に

```
1 | トロピカル無職 | 苦茶の音楽工房 | path/to/mp3
2 | 君が代 | 林廣守 | path/to/wav
となり曲名が変化した.
```

9 ORDER BYを含むSQL文

```
select players.player_name, playData.best_score
from playData
JOIN charts ON playData.chart_id = charts.chart_id
JOIN players ON playData.player_id = players.player_id
where charts.chart_id = 1 ORDER BY playData.best_score DESC
```

説明

特定の楽曲(chart_id=1)のスコアのランキングを取得するためのsql文.

実行結果

```
はるし|1000000  
ボル・ポット|798999  
スターリソ|500000  
港悪亜|0
```

10 CREATE VIEWを含むSQL文

```
CREATE VIEW user_played_songs AS  
SELECT p.player_name, s.title, s.composer_name, s.file_path  
FROM players p  
JOIN playData pd ON p.player_id = pd.player_id  
JOIN charts c ON pd.chart_id = c.chart_id  
JOIN songs s ON c.song_id = s.song_id;
```

説明

特定のユーザーがプレイした全楽曲の情報を管理するためのビューを作成するためのview.

実行結果

出力なし