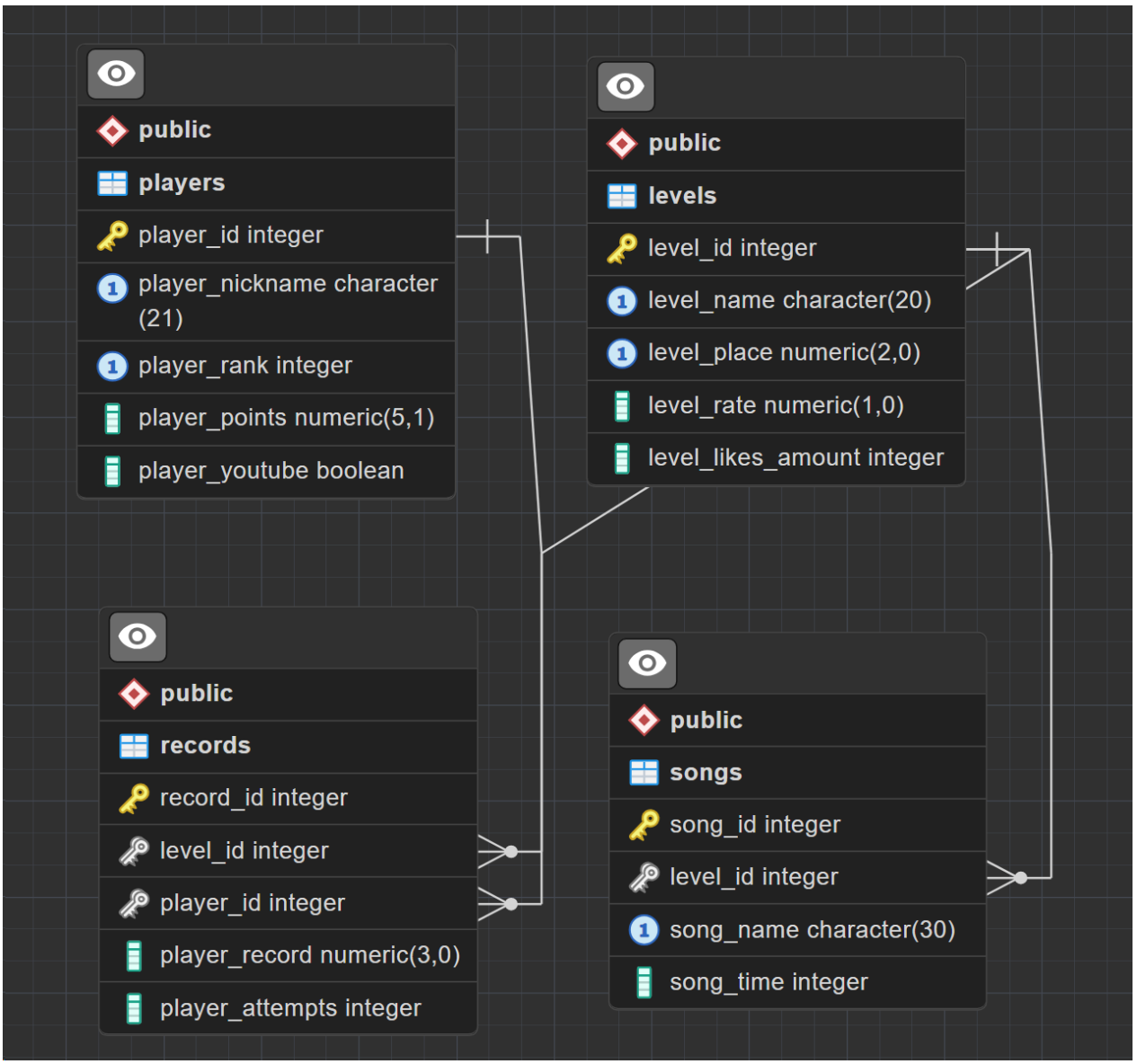
Курсовая Работа
по Базам Данных

Тема: БД по игре Geometry Dash
Подготовил: Козлов Сергей Алексеевич, 22.Б08
Преподаватель: Митрофанов Евгений Павлович
Университет: СПбГУ

Предметная область БД:
База данных представляет собой совокупность данных об игроках и уровнях игры Geometry Dash. С помощью нее удобно составлять рейтинг игроков и сравнивать достижения. База устроена следующим образом:



База данных содержит таблицы:

1. players
   Главная таблица по отношению к таблицам records, которая содержит информацию о нике игрока, место игрока в рейтинге, его очки и наличие YouTube канала.

2. .levels
   Главная таблица по отношению к таблицам records и songs. Представляет собой информацию об уровнях.

3. records
   Таблица, которая является совокупностью данных из players и levels. Игрокам из players и уровням из levels ставится в соответствие рекорд и количество попыток.

4. songs
   Информация о музыке уровня.

Отношения:

- 1 : m
  У разных уровней может быть одна и та же музыка.

- m : m
  Разные игроки могут проходить разные уровни.

- 1 : 1
  Одному рекорду соответствует один игрок и уровень.

Запросы

Простые запросы:

1. Вывод всех уровней с оценкой либо 2, либо 3. Сортируем по убыванию оценки, затем по убыванию лайков на уровне(оптимизируем при помощи levels_level_rate_index):

```
SELECT level_name, level_rate, level_likes_amount FROM levels
WHERE (level_rate = 2 OR level_rate = 3)
ORDER BY level_rate DESC, level_likes_amount DESC;
```

2. Вывод топ 5 игроков. Сортируем по убыванию ранга, начиная с 1-ого(оптимизируем при помощи players_player_rank_index):

```
SELECT player_nickname, player_rank FROM players
WHERE player_rank <= 5
ORDER BY player_rank;
```

3. Вывод песен, с длительностью не менее двух и меньше трех(оптимизируем при помощи songs_song_time_index):

```
SELECT song_name, song_time FROM songs
WHERE song_time / 60 = 2
ORDER BY song_name;
```

4.Выводим айди игроков, количество их пройденных уровней, общее и среднее количество попыток:

```sql
SELECT player_id,
    COUNT(level_id) AS levels_count,
    SUM(player_attempts) AS attempts,
    ROUND(SUM(player_attempts) / COUNT(level_id), 2) AS avg_attempts
FROM records
GROUP BY player_id
ORDER BY ROUND(SUM(player_attempts) / COUNT(level_id), 2);
```

Средние запросы:

1. Выводим топ 5 игроков по наименьшему количеству попыток в среднем:

```sql
SELECT player_nickname, ROUND(AVG(player_attempts), 0) AS avg_attempts FROM
    players RIGHT JOIN records USING(player_id)
GROUP BY player_nickname
ORDER BY ROUND(AVG(player_attempts), 0)
LIMIT 5;
```

2. Выводим level_name и song_name, при условии, что длина данной песни от 130 секунд до 210:

```sql
SELECT level_name, song_name FROM
    levels INNER JOIN songs USING(level_id)
WHERE song_time BETWEEN 130 AND 210
ORDER BY song_time DESC;
```

3. Выводим никнеймы игроков, которые присутствуют в таблицу рекордов:

```sql
SELECT player_nickname, player_points FROM
    players LEFT JOIN records USING(player_id)
WHERE player_attempts IS NULL
ORDER BY player_points DESC;
```

Сложные запросы:

1. Выводим название уровней, их песни, количество игроков, рекорд которых зафиксирован на уровне и у которых есть YouTube канал, и количество лайков на уровне. Сортируем сначала по количеству игроков, игравших в уровень, затем по количеству лайков:

```
SELECT level_name, song_name, COUNT(player_id), SUM(level_likes_amount) FROM
    levels
    RIGHT JOIN records USING(level_id)
    INNER JOIN songs USING(level_id)
WHERE records.player_id IN (
    SELECT player_id FROM players WHERE player_youtube = TRUE
)
GROUP BY level_name, song_name
ORDER BY COUNT(player_id) DESC, SUM(level_likes_amount) DESC;
```

2. Выводим ник игрока, его рекорд, попытки на уровень и название уровня, длина песни которого больше 170 секунд:

```
SELECT level_name, player_nickname, player_record, player_attempts FROM
    levels
    RIGHT JOIN records USING(level_id)
    LEFT JOIN players USING(player_id)
WHERE records.level_id IN (
    SELECT levels.level_id FROM
        levels INNER JOIN songs USinG(level_id)
    WHERE song_time > 170
)
ORDER BY player_record, player_attempts;
```

3. Выводим ник игрока, название уровня, называние музыки и рекорд игрока на уровне при условии,что общее количество попыток игроков на всех уровнях больше 100000:

```sql
SELECT player_nickname, level_name, song_name, player_record FROM
    levels
    RIGHT JOIN records USING(level_id)
    INNER JOIN songs USING(level_id)
    LEFT JOIN players USING(player_id)
WHERE player_nickname IN (
    SELECT player_nickname FROM
        players INNER JOIN records USING(player_id)
    GROUP BY player_nickname HAVING SUM(player_attempts) > 100000
)
ORDER BY player_record DESC, level_name ASC;
```