

Комбинация данных из нескольких таблиц

Соединение

- операция реляционной базы данных, комбинирующая поля из одной и более таблиц в едином результате запроса.

Соединение происходит, когда после FROM в запросе SELECT перечисляется более чем одно имя таблицы.

```
SELECT MOVIE_ID, MOVIE_GENRE_DESCRIPTION AS  
GENRE, MOVIE_TITLE  
FROM MOVIE, MOVIE_GENRE  
ORDER BY MOVIE_ID;
```

Ответ: 16жанров*20фильмов=320 строк

Эквисоединение

(внутреннее соединение) – соединение, в котором одно или более полей одной таблицы (внешний ключ) совмещаются с аналогичными полями другой таблицы (первичный ключ), при условии равенства (т.е. совпадают, если данные в них одинаковы)

Задание 1

Выведите номер MOVIE_ID, название фильма MOVIE_TITLE из таблицы MOVIE и соответствующее описание жанра из MOVIE_GENRE следующими способами:

- А) Для соединения таблиц использовать ключевое слово WHERE;
- Б) Для соединения таблиц использовать псевдонимы таблиц.

Соединения с использованием WHERE

A)

```
SELECT MOVIE_ID, MOVIE_GENRE_DESCRIPTION AS  
GENRE, MOVIE_TITLE  
FROM MOVIE, MOVIE_GENRE  
WHERE MOVIE_GENRE.MOVIE_GENRE_CODE =  
MOVIE.MOVIE_GENRE_CODE;
```

С добавленными псевдонимами таблиц

Б)

```
SELECT A.MOVIE_ID, B.MOVIE_GENRE_DESCRIPTION  
AS GENRE, A.MOVIE_TITLE  
FROM MOVIE A, MOVIE_GENRE B  
WHERE A.MOVIE_GENRE_CODE =  
B.MOVIE_GENRE_CODE;
```


Соединения с

использованием JOIN

Оператор JOIN объединяет список таблиц после FROM и условия соединения в единую конструкцию.

имя_таблицы [INNER] JOIN имя_таблицы

{ ON условие | USING(имя_поля [,имя_поля]) }

ON позволяет указать условие, подобное указанному после WHERE;

USING перечисляет имена полей, ктр должны использоваться для совпадающих строк. Работает только тогда, когда столбцы в обеих таблицах имеют одинаковые имена (Oracle, MySQL).

Соединение с условием ON

```
SELECT MOVIE_ID,  
MOVIE_GENRE_DESCRIPTION AS GENRE,  
MOVIE_TITLE  
FROM MOVIE INNER JOIN MOVIE_GENRE  
ON  
MOVIE.MOVIE_GENRE_CODE =  
MOVIE_GENRE.MOVIE_GENRE_CODE;
```

Соединение с псевдонимами вместо имен таблиц

```
SELECT A.MOVIE_ID, B.MOVIE_GENRE_DESCRIPTION  
AS GENRE, A.MOVIE_TITLE  
FROM MOVIE AS A INNER JOIN MOVIE_GENRE AS B ON  
A.MOVIE_GENRE_CODE = B.MOVIE_GENRE_CODE;
```

Соединение с ключевым словом USING (вместо условия ON)

```
SELECT MOVIE_ID, MOVIE_GENRE_DESCRIPTION AS  
GENRE, MOVIE_TITLE  
FROM MOVIE INNER JOIN MOVIE_GENRE USING  
(MOVIE_GENRE_CODE);
```

Естественное соединение

основывается на всех полях двух таблиц и более таблиц, имена которых совпадают.

```
SELECT A.MOVIE_ID, B.MOVIE_GENRE_DESCRIPTION  
AS GENRE,  
    C.MPAA_RATING_CODE AS RATING,  
    C.MPAA_RATING_DESCRIPTION AS RATING_DESC  
FROM MOVIE A INNER JOIN MOVIE_GENRE B ON  
A.MOVIE_GENRE_CODE = B.MOVIE_GENRE_CODE  
    INNER JOIN MPAA_RATING_CODE ON  
A.MPAA_RATING_CODE = C.MPAA_RATING_CODE  
WHERE MOVIE.MOVIE_ID<6;
```

Внешнее соединение

(включительное соединение) включает несовпавшие строки как минимум одной из таблиц в результаты запроса.

Когда есть несовпавшие строки, все значения, выбранные из таблицы, где соответствующая строка не найдена, являются пустыми.

```
имя_таблицы {RIGHT|LEFT|FULL} [OUTER] JOIN  
имя_таблицы  
{ ON условие|USING (имя_поля [,имя_поля]) }
```

Задание 4

Вывести список всех описаний жанров вместе с фильмами, входящими в жанр. Отметьте строки в списке результатов, не имеющие значения для MOVIE_TITLE, - это жанры, для которых нет соответствующих фильмов.


```
SELECT MOVIE_GENRE_DESCRIPTION AS GENRE,  
MOVIE_TITLE  
FROM MOVIE RIGHT OUTER JOIN MOVIE_GENRE  
ON  
MOVIE.MOVIE_GENRE_CODE=MOVIE_GENRE.MOVIE_G  
ENRE_CODE;
```

Задание 5

Вывести список описаний жанров, для которых нет соответствующих фильмов.

```
SELECT MOVIE_GENRE_DESCRIPTION AS GENRE
FROM MOVIE RIGHT OUTER JOIN MOVIE_GENRE
  ON MOVIE.MOVIE_GENRE_CODE =
MOVIE_GENRE.MOVIE_GENRE_CODE
WHERE MOVIE_TITLE IS NULL;
```

Рефлексивное соединение

- соединение таблицы с самой собой.

Есть отношения, первичный и внешний ключ которых находятся в одной таблице.

Это называется *рекурсивной ссылкой*, такая в базе данных только одна.

Вложенные запросы

представляют собой подчиненные запросы `SELECT`.

Они обычно помещаются после `WHERE` в качестве способа ограничения строк, возвращаемых в списке результатов внешнего запроса.

Некоррелированный вложенный запрос

-вложенный запрос, в котором внутренний запрос не обращается к содержащему его внешнему.

Это означает, что сначала можно выполнить внутренний запрос, а затем использовать полученный список результатов во внешнем запросе.

Задание 6

Вывести список всех языков, на которых в видеопрокате нет фильмов.

```
SELECT LANGUAGE_CODE, LANGUAGE_NAME  
FROM LANGUAGE  
WHERE LANGUAGE_CODE NOT IN  
  (SELECT DISTINCT LANGUAGE_CODE  
   FROM MOVIE_LANGUAGE);
```

Коррелированный вложенный запрос

-вложенный запрос, в котором внутренний запрос обращается к значениям, полученным с помощью внешнего.