

## Лабораторная работа №3. Запросы на выборку. Представления.

Базы данных.

3 курс. 5 группа.

Кушинов А.В. 2023-2024 г.

Уровень сложности: **Средний**

Формат работы: Индивидуальная по вариантам.

Срок выполнения: **2 недели.**

### Цель работы

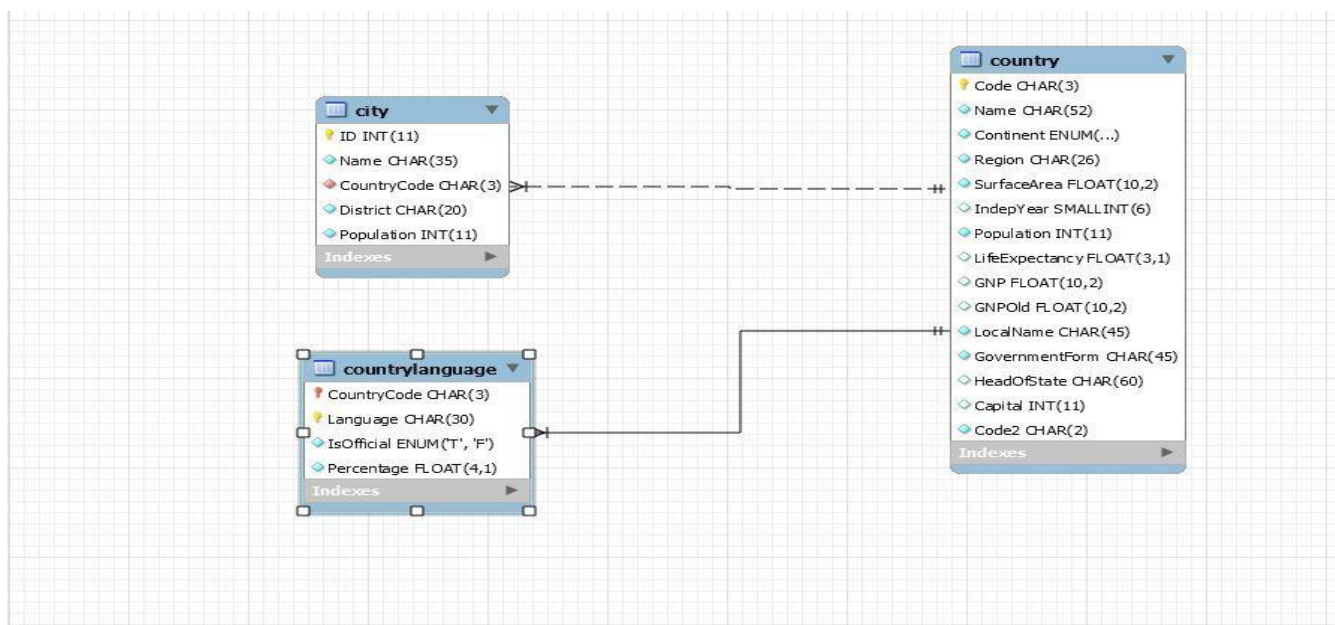
Получение навыков написания структурированных запросов на выборку, в том числе запросов на агрегацию и группировку, вложенных запросов. Знакомство с механизмом представлений.

### Минимальные теоретические сведения

Основной оператор DML – оператор SELECT. Он служит для построения запросов на выборку данных из таблиц БД. Общий вид такого оператора:

```
SELECT [DISTINCT | ALL] { * | [columnExpression [AS newName]] [, ...] }  
FROM TableName [alias] [, ...]  
[WHERE condition]  
[GROUP BY columnList] [HAVING condition]  
[ORDER BY columnList]
```

Все ключевые слова служат для достижения определённых целей. Рассмотрим примеры. Все примеры построены на базе БД world, схема которой представлена ниже.



**Пример 1:** Рассмотрим несколько элементарных запросов.

Для вывода всех записей таблицы используют уже известный вам синтаксис.

```
SELECT * FROM city;
```

Такой запрос выведет все поля всех кортежей таблицы city. Как правило, пользователю необходимо осуществить доступ лишь к ограниченному числу записей. Для этого используют оператор WHERE.

```
SELECT * FROM city
WHERE population > 1000000;
```

```
SELECT Name, Population, GovernmentForm
FROM country
WHERE Continent = 'Europe';
```

Выводим все поля записей таблицы с городами, в которых поле population > 1000000. Такой запрос называют *горизонтальной выборкой*. Во втором запросе, мы выводим не все поля таблицы со странами. Их указывают после ключевого слова SELECT, тогда выборка называется *вертикальной*.

Логическими операторами (<,>,...) можно легко манипулировать, используя операции вроде «логического и» и «логического или».

```
SELECT Name, Population, SurfaceArea
FROM country
WHERE Continent = 'Europe' AND Population > 10000000;
```

```
SELECT Name, Population, SurfaceArea, GNP
FROM country
WHERE (Continent = 'Europe' AND Population > 10000000) OR GNP > 30000;
```

Обратите внимание на следующие операторы. Ключевое слово BETWEEN позволяет задать диапазон, это что-то вроде двойного неравенства. Оператор IN определяет строки, значения которых, в заданном столбце, равны одному из значений диапазона.

```
SELECT * FROM city
WHERE population BETWEEN 1000000 AND 2000000;
```

```
SELECT * FROM city
WHERE CountryCode IN ('BLR', 'RUS') AND Population > 1000000;
```

```
SELECT * FROM city
WHERE CountryCode NOT IN ('BLR', 'RUS') AND Population > 1000000
ORDER BY Population ASC;
```

Оператор ORDER BY отвечает за сортировку выборки по определённому столбцу.

**Пример 2:** Отдельного внимания заслуживает оператор LIKE. Его стихия – строковые данные. Для нахождения строки определённого вида строят специальные шаблоны. Символ % - означает некоторое количество символов, в свою очередь, символ \_ - обозначает строго 1 символ. Итак, имеем:

```
SELECT * FROM city
WHERE Name LIKE 'A%';
```

```
SELECT * FROM city
WHERE Name LIKE '%a';
```

```
SELECT * FROM city
WHERE Name LIKE 'M%k';
```

Первый запрос, отыщет нам города, имя которых начинается с буквы «А». Второй запрос выводит города, в названии которых последняя буква «а». Третий запрос, демонстрирует ограничение на первую и последнюю буквы одновременно. В случае, когда нужно ограничить количество букв, используют \_.

```
SELECT * FROM city
WHERE Name LIKE 'R___';
```

Представленные выше запросы, позволяют получать *горизонтальные и вертикальные проекции отношений*. Однако, на практике их не достаточно – информация нуждается в дополнительной обработке и преобразовании. В частности, важнейшее место в теории запросов занимает *группировка и агрегация данных*. Агрегация данных – процесс объединения таковых в одну статистическую систему. Проще говоря, этот процесс позволяет нам извлечь из таблицы (или нескольких таблиц) некоторую статистическую информацию. В таблице представлены основные агрегирующие функции MySQL.

| Функция | Результат                       |
|---------|---------------------------------|
| COUNT() | Подсчёт количества записей      |
| MAX()   | Максимальное значение диапазона |
| MIN()   | Минимальное значение диапазона  |
| AVG()   | Среднее значение диапазона.     |
| SUM()   | Суммирует значения диапазона    |

**Пример 3:** Следующие примеры иллюстрируют применение группировки данных и агрегации.

```
SELECT Continent FROM country;
```

```
SELECT DISTINCT Continent FROM country;
```

```
SELECT COUNT(Continent) FROM country;
```

```
SELECT COUNT(DISTINCT Continent), COUNT(Continent) FROM country;
```

Подсчёт количества записей – ключевая функция. Ключевое слово DISTINCT позволяет не учитывать в запрос повторяющиеся значения. Обратите внимание на вывод последнего запроса.

| Result Grid               | Filter Rows:     |
|---------------------------|------------------|
| COUNT(DISTINCT Continent) | COUNT(Continent) |
| 7                         | 239              |

Ключевое слово DISTINCT позволяет получить реальную информацию о количестве континентов. Используйте это приём при подсчёте минимального множества повторяющихся значений. Несколько примеров представлены в таблице.

| <b>Запрос</b><br><code>SELECT Continent, COUNT(Code) AS Qua FROM country GROUP BY Continent;</code> | <b>Суть</b><br>Количество стран на каждом континенте.        | <b>Результат</b> <table><tr><th></th><th>Continent</th><th>Qua</th></tr><tr><td></td><td>Asia</td><td>51</td></tr><tr><td></td><td>Europe</td><td>46</td></tr><tr><td></td><td>North America</td><td>37</td></tr><tr><td></td><td>Africa</td><td>58</td></tr><tr><td></td><td>Oceania</td><td>28</td></tr><tr><td></td><td>Antarctica</td><td>5</td></tr><tr><td></td><td>South America</td><td>14</td></tr></table>                             |  | Continent | Qua    |  | Asia | 51         |  | Europe | 46        |  | North America | 37        |  | Africa | 58        |  | Oceania | 28       |  | Antarctica | 5 |  | South America | 14        |
|---|--|--|--|-----------|--------|--|------|------------|--|--------|-----------|--|---------------|-----------|--|--------|-----------|--|---------|----------|--|------------|---|--|---------------|-----------|
|   | Continent  | Qua  |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | Asia   | 51   |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | Europe   | 46   |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | North America  | 37   |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | Africa   | 58   |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | Oceania  | 28   |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | Antarctica   | 5  |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | South America  | 14   |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
| <code>SELECT Continent, MAX(Population) AS maxPop FROM country GROUP BY Continent;</code>           | Максимальное значение населения страны на каждом континенте. | <table><tr><th></th><th>Continent</th><th>maxPop</th></tr><tr><td></td><td>Asia</td><td>1277558000</td></tr><tr><td></td><td>Europe</td><td>146934000</td></tr><tr><td></td><td>North America</td><td>278357000</td></tr><tr><td></td><td>Africa</td><td>111506000</td></tr><tr><td></td><td>Oceania</td><td>18886000</td></tr><tr><td></td><td>Antarctica</td><td>0</td></tr><tr><td></td><td>South America</td><td>170115000</td></tr></table> |  | Continent | maxPop |  | Asia | 1277558000 |  | Europe | 146934000 |  | North America | 278357000 |  | Africa | 111506000 |  | Oceania | 18886000 |  | Antarctica | 0 |  | South America | 170115000 |
|   | Continent  | maxPop   |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | Asia   | 1277558000   |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | Europe   | 146934000  |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | North America  | 278357000  |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | Africa   | 111506000  |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | Oceania  | 18886000   |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | Antarctica   | 0  |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |
|   | South America  | 170115000  |  |           |        |  |      |            |  |        |           |  |               |           |  |        |           |  |         |          |  |            |   |  |               |           |



| <pre>SELECT Continent, AVG(Population) AS maxPop FROM country GROUP BY Continent;</pre> | Среднее население в стране по каждому континенту. | <table><tr><th>Continent</th><th>maxPop</th></tr><tr><td>Asia</td><td>72647562.7451</td></tr><tr><td>Europe</td><td>15871186.9565</td></tr><tr><td>North America</td><td>13053864.8649</td></tr><tr><td>Africa</td><td>13525431.0345</td></tr><tr><td>Oceania</td><td>1085755.3571</td></tr><tr><td>Antarctica</td><td>0.0000</td></tr><tr><td>South America</td><td>24698571.4286</td></tr></table> | Continent | maxPop | Asia | 72647562.7451 | Europe | 15871186.9565 | North America | 13053864.8649 | Africa | 13525431.0345 | Oceania | 1085755.3571 | Antarctica | 0.0000 | South America | 24698571.4286 |
|---|---|--|-----------|--------|------|---------------|--------|---------------|---------------|---------------|--------|---------------|---------|--------------|------------|--------|---------------|---------------|
| Continent   | maxPop  |  |           |        |      |               |        |               |               |               |        |               |         |              |            |        |               |               |
| Asia  | 72647562.7451                                     |  |           |        |      |               |        |               |               |               |        |               |         |              |            |        |               |               |
| Europe  | 15871186.9565                                     |  |           |        |      |               |        |               |               |               |        |               |         |              |            |        |               |               |
| North America   | 13053864.8649                                     |  |           |        |      |               |        |               |               |               |        |               |         |              |            |        |               |               |
| Africa  | 13525431.0345                                     |  |           |        |      |               |        |               |               |               |        |               |         |              |            |        |               |               |
| Oceania   | 1085755.3571                                      |  |           |        |      |               |        |               |               |               |        |               |         |              |            |        |               |               |
| Antarctica  | 0.0000  |  |           |        |      |               |        |               |               |               |        |               |         |              |            |        |               |               |
| South America   | 24698571.4286                                     |  |           |        |      |               |        |               |               |               |        |               |         |              |            |        |               |               |

Общий принцип вырисовывается. Ключевое слово GROUP BY сгруппирует ваши записи по значению некоторого поля или полей. Важно помнить, что **агрегатные функции нельзя использовать в блоке WHERE**. Если необходима дополнительная выборка, воспользуйтесь оператором HAVING.

```
SELECT Continent, MAX(Population) AS maxPop FROM country
GROUP BY Continent
HAVING maxPop>0;
```

Нормализованная многотабличная структура БД принуждает нас изучать технологии многотабличных запросов. Обойтись однотабличными запросами в современной РБД нереально.

**Пример 4:** Для соединения таблиц по внешнему ключу (или другим полям) используют оператор INNER JOIN. Например. Выведем столицы всех государств с ВВП превосходящим 30000.

```
SELECT country.Name, city.Name, city.Population
FROM country
INNER JOIN city
ON country.Capital = city.ID
WHERE country.GNP>30000;
```

Также попробуем вывести все города и соответствующие им страны Европы в одной результирующей таблице.

```
SELECT country.Name, city.Name, city.Population
FROM country
INNER JOIN city
ON country.Code = city.CountryCode
WHERE country.Continent='Europe';
```

Ключевой момент состоит в определении столбцов, по которым связаны таблицы. Они указаны после слова ON. Именно одинаковые значения в этих столбцах позволяют системе объединить связанные записи в результат. Вообще говоря, для получения схожих результатов можно использовать оператор WHERE. Следующий запрос аналогичен предыдущему по результату.

```
SELECT country.Name, city.Name, city.Population
FROM country, city
WHERE country.Code = city.CountryCode
AND country.Continent='Europe';
```

Само собой, можем «склеить» и три и более таблиц.

```
SELECT country.Name, city.Name,
city.Population, countrylanguage.Language
FROM country
INNER JOIN city
ON country.Capital = city.ID
INNER JOIN countrylanguage
ON country.Code = countrylanguage.CountryCode
WHERE country.Continent='Europe';
```

Существуют и другие операторы для объединения таблиц, такие как LEFT JOIN, CROSS JOIN, RIGHT JOIN, NATURAL JOIN. Информацию о них ищите в лекционном материале.

**Пример 5:** Особое место занимает теория вложенных запросов. Язык SQL – позволяет встраивать запросы в другие запросы.

```
SELECT country.Name, country.Population
FROM country,
(
  SELECT country.Population AS pop FROM country
  WHERE country.Name='Belarus'
) AS res
WHERE country.Population BETWEEN res.pop-100000
AND res.pop+100000;
```

После ключевого слова FROM мы, помимо таблиц, можем указать и вложенный запрос, ведь он по сути тоже таблица, хоть и временная. Обозначив запрос псевдонимом с помощью AS, получаем возможность обращаться к нему в основном запросе. В запросе выше получаем страны, население которых не отстоит от населения Беларуси более чем на 100000.

```
SELECT country.Name, tb1.*
FROM country,
(
  SELECT Continent, MAX(Population) AS maxPop
  FROM country
  GROUP BY Continent
) as tb1
WHERE country.Population=tb1.maxPop;
```

Вложенные запросы вполне можно использовать с запросами на группировку и с операциями над множествами.

```
(SELECT city.* FROM city
INNER JOIN country
ON country.Code=city.CountryCode
WHERE country.Name = 'Belarus')
UNION
(SELECT city.* FROM city
INNER JOIN country
ON country.Code=city.CountryCode
WHERE country.Name = 'Ukraine');
```

*Представление* – это запрос на выборку, сохранённый в базе данных.

Представление позволяет увидеть результаты сохранённого запроса таким образом, как будто это полноценная таблица БД.

Синтаксис создания представления:

```
CREATE
  [OR REPLACE]
  [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
  [DEFINER = { user | CURRENT_USER }]
  [SQL SECURITY { DEFINER | INVOKER }]
  VIEW view_name [(column_list)]
  AS select_statement
  [WITH [CASCADED | LOCAL] CHECK OPTION]
```

По сути, сведений для создания запросов достаточно и для создания представлений. Приведём несколько примеров.

**Пример 6:** Создадим представление БД world для Ангелы Меркель.

```
CREATE OR REPLACE VIEW economical_merkel
(Country, VNP, VNP_na_dushu_naselenia) AS
SELECT Name, Population, GNP/Population
FROM country
WHERE country.Continent='Europe';
```

Обращаться к нему можем также, как к любой таблице.

```
SELECT * FROM economical_merkel
WHERE VNP<10000;
```

**Пример 7:** Самостоятельно изучите результат следующего запроса. Он использует оконную функцию Rank(). Подумайте, как её можно использовать?

```
select * , Rank() over (partition by CountryCode order by population desc) as r from city;
```

### Задания для самостоятельной работы

1. Импортируйте на свою машину учебные БД aero и world. Изучите их структуру.
2. Выполните SQL – запросы согласно варианту.

| Вар | Запросы.   |
|-----|--|
| 1   | <ol style="list-style-type: none"> <li>1. Выведите страны, в которых ВВП превышает 20000.</li> <li>2. Сколько стран мира используют русский язык?</li> <li>3. Найдите страны по площади, превышающие территорию самой большой страны в Африке.</li> <li>4. Найдите города, которые по населению превышают <b>ровно 3</b> европейских государства.</li> <li>5. Выведите страны, среднее население в городах которой превышает население в столице.</li> </ol>   |
| 2   | <ol style="list-style-type: none"> <li>1. Выведите список всех стран и столиц.</li> <li>2. Какой средний процент использования английского языка по странам мира?</li> <li>3. Сконструируйте запрос, выводящий все поездки в Москву с указанием имени пассажира.</li> <li>4. Найдите страны, которые по суммарному количеству владеющих официальными языками превосходят суммарное население двух самых крупных городов.</li> <li>5. Вывести города, которые в своём названии имеют два разных слова и располагаются в странах со значением ВВП, превосходящее среднее по миру.</li> </ol> |
| 3   | <ol style="list-style-type: none"> <li>1. Вывести все города, которые являются столицами какого-либо государства.</li> <li>2. Найдите все страны, в которых используют 1 язык.</li> <li>3. Сколько всего городов сравнимых с Минском расположено в Африке?</li> <li>4. Найдите города Северной Америки, которые по населению превышают ровно 4 азиатских города.</li> <li>5. Выведите страны, в которых количество используемых языков ниже среднемировых показателей.</li> </ol>  |
| 4   | <ol style="list-style-type: none"> <li>1. Вывести все языки, которые используются в нескольких странах одновременно.</li> <li>2. Выведите страны Северной Америки вместе с языками, которые используют в этих странах.</li> <li>3. Выведите список стран, которые не имеют городов.</li> <li>4. Выведите города, первые две буквы названия которых совпадают с первыми двумя буквами названия страны, в которых они расположены.</li> </ol>  |

|    |  |
|----|--|
|    | 5. Выведите минимальные показатели ВВП по континентам с указанием стран, которые их выдают.  |
| 5  | <ol style="list-style-type: none"> <li>1. Найдите страны, получившие независимость позже Республики Беларусь.</li> <li>2. Выведите страны Африки вместе с языками, которые используют в этих странах.</li> <li>3. Найдите среднее население в городах каждой страны.</li> <li>4. Найдите страны, в которых процент владения официальными языками не превышает процент владения неофициальными.</li> <li>5. Вывести языки, которые используются в Африке и при этом на них говорит суммарно большее количество населения чем в 3 самых крупных городах Европы.</li> </ol> |
| 6  | <ol style="list-style-type: none"> <li>1. Найдите страны, в которых код страны и название страны начинаются и заканчиваются одинаково.</li> <li>2. Сколько стран мира имеют ВВП превосходящий суммарный ВВП РБ и Украины.</li> <li>3. Выведите города, занимающие по населению в мире позиции с 200 по 2034.</li> <li>4. Найдите количество городов в каждом регионе Европы.</li> <li>5. В каком полушарии (деление может быть условным) проживает больше людей? Выведите также процентное распределение населения Земли по континентам.</li> </ol>                      |
| 7  | <ol style="list-style-type: none"> <li>1. Выведите все города мира, в которых можно услышать французский язык.</li> <li>2. Какой процент населения планеты проживает в странах с ВВП ниже 10000?</li> <li>3. Выведите языки мира, занимающие по числу стран использования в мире позиции с 20 по 27.</li> <li>4. Сколько городов Украины имеют население превосходящее число людей, говорящих в Украине на русском.</li> <li>5. Выведите города России, в которых проживает не более 100000 человек.</li> </ol>  |
| 8  | <ol style="list-style-type: none"> <li>1. Сколько городов с населением выше 1000000 человек находится на каждом континенте.</li> <li>2. Выведите страны Азии, среднее население в которых превышает среднее население в первых 30 городах России по населению.</li> <li>3. Сколько всего различных языков можно встретить в странах с различными формами правления?</li> <li>4. Сколько всего регионов в Европе?</li> <li>5. Вывести страны количество чётных цифр в году независимости которых, превышает количество официальных языков.</li> </ol>                     |
| 9  | <ol style="list-style-type: none"> <li>1. Вывести страны мира, которые получили независимость в високосный год.</li> <li>2. Какой процент населения мира проживает в странах, которые получили независимость после 1950 года?</li> <li>3. Выведите страны Северной Америки, среднее население в которых превышает среднее население в первых 30 городах России по населению.</li> <li>4. Выведите среднее население на каждом континенте.</li> <li>5. Выведите страны в которых на официальном языке говорит больше людей, чем живёт в столице.</li> </ol>               |
| 10 | <ol style="list-style-type: none"> <li>1. Найдите страны, названия которых начинаются с буквы «А».</li> <li>2. Выведите все города мира, в которых можно услышать русский язык.</li> <li>3. Выведите страны, среднее население в которых не превышает население в столице.</li> <li>4. Найдите количество городов в каждом регионе Азии.</li> <li>5. Вывести страны, в которых на неофициальных языках разговаривает больше людей, чем среднее население в 5 самых крупных городах страны. Названия этих городов вывести через запятую в отдельной ячейке.</li> </ol>    |

- |  |  |
|--|--|
|  |  |
|--|--|
- a. Выгрузите в файл результаты последнего запроса.
  3. Создайте несколько учебных запросов к вашей БД из ЛР1. Среди них.
    - a. Запрос к 4 –ем таблицам одновременно.
    - b. 5 запросов на группировку.
    - c. 3 вложенных запроса.
    - d. Запрос с использованием операций над множествами.
    - e. Обновление таблиц с использованием оператора соединения.
    - f. Запрос с использованием оконных функций.
  4. Создайте представления для различных участников проекта из ЛР1. (Администратора, продавца, рабочего...). Преобразуйте данные необходимым образом. Учитывайте уровень и степень доступа к данным.

Литература:

1. Кузнецов, Симдянов – MySQL 5.0.
2. Линн Бейли – Изучаем SQL.
3. Кронке – Теория и практика построения баз данных.
4. Коннолли, Берг – Базы данных.