**Лабораторная работа № 7**

Цель: научится создавать и использовать представления в СУБД MySQL.

Задачи:

1. Изучить понятие VIEW;

2. Изучить и знать наизусть синтаксис;

3. Создать представление в базе данных лабораторной работы № 5;

4. Реализовать представления на базе запросов 2 лабораторной работы, либо 5 ЛБ в количестве 3 штук;

5. Проверить работу представлений, в коде реализуемой программы;

6. Отчет оформить, используя шаблон отчета.

***Срок выполнения лабораторной работы 1 неделя***

**Теория**

Представление (VIEW) — объект базы данных, являющийся результатом выполнения запроса к базе данных, определенного с помощью оператора SELECT, в момент обращения к представлению.

Представления иногда называют «виртуальными таблицами». Такое название связано с тем, что представление доступно для пользователя как таблица, но само оно не содержит данных, а извлекает их из таблиц в момент обращения к нему. Если данные изменены в базовой таблице, то пользователь получит актуальные данные при обращении к представлению, использующему данную таблицу; кэширования результатов выборки из таблицы при работе представлений не производится. При этом, механизм кэширования запросов (query cache) работает на уровне запросов пользователя безотносительно к тому, обращается ли пользователь к таблицам или представлениям.

Синтаксис создания представления в MySQL:

CREATE [OR REPLACE]  
[ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]  
VIEW view\_name [(column\_list)]  
AS select\_statement  
[WITH [CASCADED | LOCAL] CHECK OPTION]

Оператор CREATE VIEW содержит 4 необязательные конструкции:

1. OR REPLACE — при использовании данной конструкции в случае существования представления с таким именем старое будет удалено, а новое создано. В противном случае возникнет ошибка, информирующая о сществовании представления с таким именем и новое представление создано не будет. Следует отметить одну особенность — имена таблиц и представлений в рамках одной базы данных должны быть уникальны, т.е. нельзя создать представление с именем уже существующей таблицы. Однако конструкция OR REPLACE действует только на представления и замещать таблицу не будет.
2. ALGORITM — определяет алгоритм, используемый при обращении к представлению (подробнее речь об этом пойдет ниже).
3. column\_list — задает имена полей представления.
4. WITH CHECK OPTION — при использовании данной конструкции все добавляемые или изменяемые строки будут проверяться на соответствие определению представления. В случае несоответствия данное изменение не будет выполнено. Обратите внимание, что при указании данной конструкции для необновляемого представления возникнет ошибка и представление не будет создано. (подробнее речь об этом пойдет ниже).