Урок 2. SQL – создание объектов, простые запросы выборки

1. Используя операторы языка SQL, создайте табличку “sales”. Заполните ее данными.
2. Для данных таблицы “sales” укажите тип заказа в зависимости от кол-ва : меньше 100 - Маленький заказ; от 100 до 300 - Средний заказ; больше 300 - Большой заказ.
3. Создайте таблицу “orders”, заполните ее значениями. Выберите все заказы. В зависимости от поля order\_status выведите столбец full\_order\_status: OPEN – «Order is in open state» ; CLOSED - «Order is closed»; CANCELLED - «Order is cancelled»
4. Чем NULL отличается от 0?

*Данные для таблиц “sales” и “orders” размещены на слайде "домашнее задание" презентации "Семинар №2. Базы данных и SQL"*

### Урок 3. SQL – выборка данных, сортировка, агрегатные функции

**Работаем с таблицей staff (скрипт создания в материалах к уроку)**

1. Отсортируйте данные по полю заработная плата (salary) в порядке: убывания; возрастания.

2. Выведите 5 максимальных заработных плат (salary).

3. Посчитайте суммарную зарплату (salary) по каждой специальности (роst).

4. Найдите кол-во сотрудников с специальностью (post) «Рабочий» в возрасте от 24 до 49 лет включительно..

5. Найдите количество специальностей.

6. Выведите специальности, у которых средний возраст сотрудников меньше 30 лет

Урок 4. SQL – работа с несколькими таблицами

1. Подсчитать общее количество лайков, которые получили пользователи младше 12 лет.
2. Определить кто больше поставил лайков (всего): мужчины или женщины.
3. Вывести всех пользователей, которые не отправляли сообщения.
4. (по желанию)\* Пусть задан некоторый пользователь. Из всех друзей этого пользователя.

Урок 5. SQL – оконные функции

**Для решения задач используйте базу данных lesson\_4** (скрипт создания, прикреплен к 4 семинару).

1. Создайте представление, в которое попадет информация о пользователях (имя, фамилия, город и пол), которые не старше 20 лет.

2. Найдите кол-во, отправленных сообщений каждым пользователем и выведите ранжированный список пользователь, указав указать имя и фамилию пользователя, количество отправленных сообщений и место в рейтинге (первое место у пользователя с максимальным количеством сообщений) . (используйте DENSE\_RANK).

3. Выберите все сообщения, отсортируйте сообщения по возрастанию даты отправления (created\_at) и найдите разницу дат отправления между соседними сообщениями, получившегося списка. (используйте LEAD или LAG).

Урок 6. SQL – Транзакции. Временные таблицы, управляющие конструкции, циклы

**Для решения задач используйте базу данных lesson4** (скрипт создания, прикреплен к 4 семинару).

1. Создайте таблицу users\_old, аналогичную таблице users. Создайте процедуру, с помощью которой можно переместить любого (одного) пользователя из таблицы users в таблицу users\_old. (использование транзакции с выбором commit или rollback – обязательно).
2. Создайте хранимую функцию hello(), которая будет возвращать приветствие, в зависимости от текущего времени суток. С 6:00 до 12:00 функция должна возвращать фразу "Доброе утро", с 12:00 до 18:00 функция должна возвращать фразу "Добрый день", с 18:00 до 00:00 — "Добрый вечер", с 00:00 до 6:00 — "Доброй ночи".