

Лабораторная работа № 4

Цель

Знакомство с основами языка SQL и построением SQL-запросов.

Задание №1

Создать базу данных для предметной области по вариантам. Вариант вычисляется по порядковому номеру в списке группы. Например, если ваш номер в списке группы 5, то вы выполняете 5 вариант. В варианте предложены несколько сущностей предметной области. Необходимо добавить еще 2 сущности, определить связи между сущностями и определить минимум по 4 атрибута каждой сущности. Следуйте стилю именования таблиц и атрибутов. Задайте первичные ключи в каждой таблице.

Варианты

1. Накладные, товары, товары в накладных.
2. Парикмахеры, услуги, выполненные работы.
3. Блюда, продукты, рецепты блюд.
4. Телефоны, виды услуг, оплата.
5. Квартиры, тарифы коммунальных платежей, оплата.
6. Инвентарь, клиенты, прокат инвентаря.
7. Склады, товары, товары на складах.
8. Фильмы, артисты, роли.
9. Издательства, магазины, книги.
10. Прививки, дети, сделанные прививки.
11. Студенты, нормативы по физкультуре, результаты.
12. Рейсы, пассажиры, билеты.
13. Сотрудники, выполненные проекты, участие в проектах.
14. Рецидивисты, статьи уголовного кодекса, отсидки.
15. Пьесы, театры, спектакли.
16. Соревнования, спортсмены, выступления.
17. Товары, продавцы, продажи.
18. Компьютеры, программы, установленные программы.
19. Врачи поликлиники, больные, прием больных.
20. Номера гостиницы, клиенты, бронирование.
21. Лекарства, больные, назначения.
22. Туристические маршруты, клиенты, путевки.
23. Аптеки, лекарства, наличие лекарств в аптеках.
24. Компьютеры, железо, комплектация.
25. Кинотеатры, фильмы, прокат фильмов.

Задание №2

Для базы данных из задания №1 необходимо сгенерировать ER диаграмму. Для этих целей нужно использовать IDE для вашей СУБД, например, SQL Server

Management Studio, PHPMyAdmin, MySQL Workbench, DataGrip или другие инструменты.

Задание №3

Написать SQL запросы. Оформить в виде файла lab4.sql с пронумерованными запросами (и заданием в комментарии для каждого запроса).

3.1 INSERT

- a. Без указания списка полей
`INSERT INTO table_name VALUES (value1, value2, value3, ...);`
- b. С указанием списка полей
`INSERT INTO table_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);`
- c. С чтением значения из другой таблицы
`INSERT INTO table2 (column_name(s)) SELECT column_name(s) FROM table1;`

3.2. DELETE

- a. Всех записей
- b. По условию
`DELETE FROM table_name WHERE condition;`

3.3. UPDATE

- a. Всех записей
- b. По условию обновляя один атрибут
`UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;`
- c. По условию обновляя несколько атрибутов
`UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;`

3.4. SELECT

- a. С набором извлекаемых атрибутов (`SELECT atr1, atr2 FROM...`)
- b. Со всеми атрибутами (`SELECT * FROM...`)
- c. С условием по атрибуту (`SELECT * FROM ... WHERE atr1 = value`)

3.5. SELECT ORDER BY + TOP (LIMIT)

- a. С сортировкой по возрастанию ASC + ограничение вывода количества записей
- b. С сортировкой по убыванию DESC
- c. С сортировкой по двум атрибутам + ограничение вывода количества записей
- d. С сортировкой по первому атрибуту, из списка извлекаемых

3.6. Работа с датами

Необходимо, чтобы одна из таблиц содержала атрибут с типом DATETIME. Например, таблица авторов может содержать дату рождения автора.

- a. WHERE по дате

- b. WHERE дата в диапазоне
- c. Извлечь из таблицы не всю дату, а только год. Например, год рождения автора. Для этого используется функция YEAR (<https://docs.microsoft.com/en-us/sql/t-sql/functions/year-transact-sql?view=sql-server-2017>)

3.7. Функции агрегации

- a. Посчитать количество записей в таблице
- b. Посчитать количество уникальных записей в таблице
- c. Вывести уникальные значения столбца
- d. Найти максимальное значение столбца
- e. Найти минимальное значение столбца
- f. Написать запрос COUNT() + GROUP BY

3.8. SELECT GROUP BY + HAVING

- a. Написать 3 разных запроса с использованием GROUP BY + HAVING. Для каждого запроса написать комментарий с пояснением, какую информацию извлекает запрос. Запрос должен быть осмысленным, т.е. находить информацию, которую можно использовать.

3.9. SELECT JOIN

- a. LEFT JOIN двух таблиц и WHERE по одному из атрибутов
- b. RIGHT JOIN. Получить такую же выборку, как и в 3.9 а
- c. LEFT JOIN трех таблиц + WHERE по атрибуту из каждой таблицы
- d. INNER JOIN двух таблиц

3.10. Подзапросы

- a. Написать запрос с условием WHERE IN (подзапрос)
- b. Написать запрос SELECT atr1, atr2, (подзапрос) FROM ...
- c. Написать запрос вида SELECT * FROM (подзапрос)
- d. Написать запрос вида SELECT * FROM table JOIN (подзапрос) ON ...