Цель этой лабораторной работы – научить вас создавать и использовать базу данных с помощью языка SQL. Вы изучите:

- Как устанавливать связи между таблицами с помощью ограничения \*\*FOREIGN KEY\*\*.

- Как добавлять данные в таблицы.

- Как выполнять SQL-запросы для работы с данными.

---

# \*\*СОХРАНЕНИЕ ДАННЫХ\*\*

Если вы выполняете лабораторную работу в несколько сессий, важно сохранять изменения в базе данных в конце каждой сессии. Это обеспечит сохранность данных и позволит продолжить работу с того места, где вы остановились. Для сохранения данных выполните команду:

```sql

COMMIT;

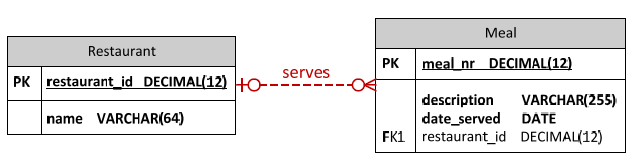
```

---

ПЕРВАЯ ЧАСТЬ

ОБЗО

В первой части работы мы будем работать с таблицами \*\*Restaurant\*\* (Ресторан) и \*\*Meal\*\* (Блюдо). Эти две таблицы связаны следующим образом:



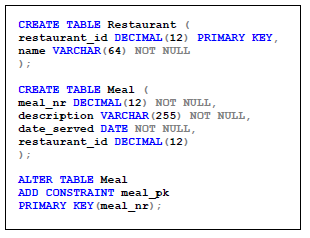
- Один \*\*ресторан\*\* может предлагать \*\*много блюд\*\*.

- Каждое \*\*блюдо\*\* предлагается только \*\*в одном ресторане\*\*.

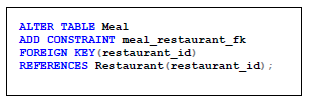
ШАГИ ВЫПОЛНЕНИЯ

1. Создайте таблицы \*\*Meal\*\* и \*\*Restaurant\*\* с помощью команды \*\*CREATE TABLE\*\*. Учитывайте форматирование команд (пробелы, скобки и переносы строк).

2. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*

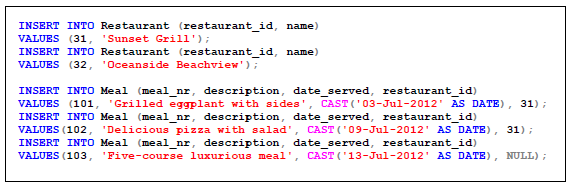


3. Создайте связь между таблицами с помощью \*\*FOREIGN KEY\*\*. В таблице \*\*Meal\*\* внешний ключ `restaurant\_id` ссылается на первичный ключ таблицы \*\*Restaurant\*\*.



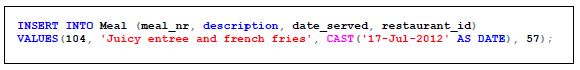
4. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

5. Вставьте тестовые данные в таблицы \*\*Restaurant\*\* и \*\*Meal\*\* с помощью команды \*\*INSERT INTO\*\*.



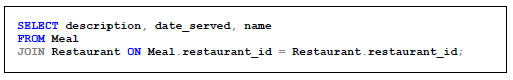
6. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

7. Проверьте, что система не позволяет вставить в таблицу \*\*Meal\*\* запись с `restaurant\_id`, которого нет в таблице \*\*Restaurant\*\*.



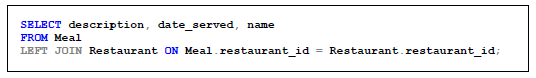
8. Сделайте \*\*скриншот ошибки\*\* и поясните, почему система отклонила команду.

9. Напишите SQL-запрос для вывода \*\*списка всех блюд, предлагаемых в ресторанах\*\*. Используйте \*\*JOIN\*\*.



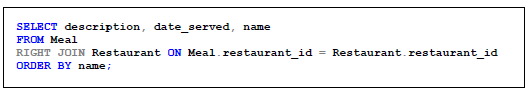
10. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

11. Напишите SQL-запрос, который покажет \*\*все блюда, даже если они не принадлежат ни одному ресторану\*\*. Используйте \*\*LEFT JOIN\*\*.



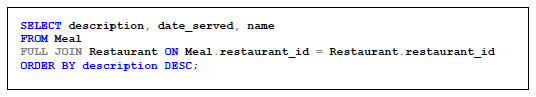
12. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

13. Напишите SQL-запрос, который покажет \*\*все рестораны, даже если они не предлагают ни одного блюда\*\*. Используйте \*\*RIGHT JOIN\*\*.



14. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

15. Напишите SQL-запрос, который покажет \*\*все блюда и все рестораны\*\*, независимо от их связи. Используйте \*\*FULL JOIN\*\*.



16. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

---

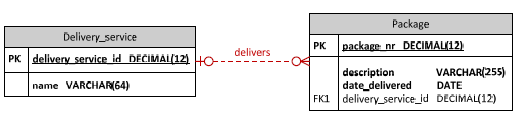
ВТОРАЯ ЧАСТЬ (Домашка)

Во второй части работы вы примените изученные навыки на новых таблицах:

- \*\*Delivery\_service\*\* (Служба доставки)

- \*\*Package\*\* (Посылка)

Связь между таблицами: \*\*Одна служба доставки может доставлять несколько посылок\*\*.



ШАГИ ВЫПОЛНЕНИЯ

1. Создайте таблицы \*\*Delivery\_service\*\* и \*\*Package\*\* с учетом всех ограничений (NOT NULL, PRIMARY KEY, FOREIGN KEY).

2. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

3. Добавьте в таблицу \*\*Delivery\_service\*\* следующие службы доставки:

- United Package Delivery

- Federal Delivery

- Dynamic Duo Delivery

4. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

5. Добавьте в таблицу \*\*Package\*\* следующие посылки:

- \*\*United Package Delivery\*\*: "Perfect diamonds", 29-Apr-2012

- \*\*Federal Delivery\*\*: "Care package", 14-Jun-2012

- \*\*Без службы доставки\*\*: "French wine", 19-Jul-2012

6. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

7. Попробуйте добавить в \*\*Package\*\* посылку с `delivery\_service\_id`, которого нет в \*\*Delivery\_service\*\*.

8. Сделайте \*\*скриншот ошибки\*\* и поясните, почему система отклонила команду.

9. Объясните:

- Почему команда из \*\*шага 7\*\* завершилась ошибкой?

- Какое сообщение об ошибке вы получили?

10. Перечислите \*\*различия\*\* между `INNER JOIN`, `LEFT JOIN`, `RIGHT JOIN` и `FULL JOIN`.

11. Напишите SQL-запрос, который покажет \*\*все посылки с указанием службы доставки\*\*.

12. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

13. Объясните, почему \*\*все строки из таблиц не отображаются\*\* при использовании `INNER JOIN`.

14. Напишите SQL-запрос, который покажет \*\*все службы доставки и их посылки (если они есть)\*\*. Используйте `LEFT JOIN`.

15. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

16. Напишите SQL-запрос, который покажет \*\*все посылки и их службы доставки (если они есть)\*\*, отсортированные по имени службы доставки.

17. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

18. Напишите SQL-запрос, который покажет \*\*все посылки и все службы доставки\*\*, независимо от их связи, с сортировкой по убыванию. Используйте `FULL JOIN`.

19. Сделайте \*\*скриншот выполнения команды\*\* и \*\*результат выполнения\*\*.

---

### \*\*ЗАКЛЮЧЕНИЕ\*\*

Вы научились:

- Создавать таблицы и устанавливать связи между ними.

- Использовать \*\*PRIMARY KEY\*\* и \*\*FOREIGN KEY\*\*.

- Выполнять \*\*JOIN\*\*-запросы для получения данных.

- Работать с ограничениями \*\*NOT NULL\*\*, \*\*FOREIGN KEY\*\* и \*\*UNIQUE\*\*.

Лабораторная работа завершена.