ЛАБОРАТОРНАЯ РАБОТА №1

**СОЗДАНИЕ И МОДИФИКАЦИЯ БАЗ ДАННЫХ**

**И ТАБЛИЦ**

**ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

# **Создание базы данных и схемы**

Создание новой базы данных производится командой

*CREATE DATABASE имя\_БД;*

PostgreSQL не позволяет выбрать активную БД с помощью запроса, поэтому сделать это можно с использованием интерфейса IDE или терминальных команд.

Удалить базу данных можно с помощью команды

*DROP DATABASE имя\_БД;*

Создать новую схему в БД можно командой

*CREATE SCHEMA имя\_схемы;*

Задать схему по умолчанию можно с помощью интерфейса IDE или запроса:

*SET SEARCH\_PATH TO имя\_схемы;*

Удаление схемы:

*DROP SCHEMA имя\_схемы;*

# **Создание и удаление таблицы**

Для создания таблицы и других объектов используется оператор **CREATE**

*CREATE TABLE [IF NOT EXISTS] имя\_таблицы (*

*поле\_1 тип\_данных,*

*[поле \_2 тип\_данных,]*

*...*

*[поле \_n тип\_данных] );*

Помимо названия столбца и его типа в определение иногда необходимо добавлять следующие необязательные параметры:

* PRIMARY KEY – первичный ключ.
* AUTO\_INCREMENT – автоматически увеличивающееся значение
* UNIQUE – значения поля не повторяются
* NOT NULL – не пустое поле
* DEFAULT – значение поля по умолчанию.

Дополнительно к таблице могут создаваться ограничения внешнего ключа, индексы и т.д.

Для удаления таблицы и других объектов используется оператор **DROP TABLE**.

*DROP TABLE [IF EXISTS] имя\_таблицы;*

# **Вставка, обновление и удаление записей в таблице**

Для добавления новых записей в таблицу предназначен оператор **INSERT**.

*INSERT INTO имя\_таблицы [(поле\_1, …, поле\_n)]*

*VALUES (значение\_1, …, значение\_n)*

Для редактирования существующих записей в таблицах используется оператор **UPDATE**.

*UPDATE имя\_таблицы*

*SET поле\_1 = значение\_1, …, поле\_n = значение\_n*

*[WHERE условие]*

Для удаления записей из таблицы используется оператор **DELETE**.

*DELETE FROM имя\_таблицы*

*[WHERE условие\_отбора\_записей];*

Если условие отбора записей WHERE отсутствует, то будут удалены все записи указанной таблицы.

# **Модификация структуры таблицы**

Структуру уже созданной таблицы можно изменить оператором **ALTER TABLE**. Эта команда имеет множество форм и может применяться для добавления и удаления полей, изменения их типов данных, значения по умолчанию, добавления и удаления ограничений и т.д.

*ALTER TABLE имя\_таблицы*

*ADD COLUMN [ IF NOT EXISTS ] полe тип\_данных;*

Аналогично можно удалить поле

*ALTER TABLE имя\_таблицы*

*DROP COLUMN [ IF EXISTS ] полe;*

Чтобы преобразовать столбец в другой тип данных, используется команда:

*ALTER TABLE имя\_таблицы*

*ALTER COLUMN полe TYPE тип\_данных;*

**Выборка записей**

Для выбора всей информации из таблицы или представления используется команда **SELECT:**

*SELECT \* FROM имя\_объекта*

Для получения данных только определенных полей (столбцов), необходимо перечислить их через запятую после ключевого слова SELECT.

*SELECT поле\_1, …, поле\_n FROM имя\_объекта*

В итоговой выборке столбцам можно назначить новое название (псевдоним)   
с помощью оператора AS (или без него).

*SELECT поле [AS] псевдоним FROM имя\_объекта*

Для того, чтобы вывести данные из столбца БЕЗ повторяющихся значений, используется параметр **DISTINCT**.

*SELECT DISTINCT поле FROM имя\_объекта*

Выбор требуемых записей (строк) осуществляется применением фразы   
**WHERE <условие>.**

Условие является логическим оператором, в котором могут использоваться операторы сравнения =, >, <, >=, <=, !=, <> и логические операторы AND, OR, NOT.

*SELECT \* FROM имя\_объекта  
WHERE поле\_1 = значение\_1 AND поле\_2 > значение\_2*

Оператор IN определяет набор значений, в который данное значение может быть включено.

*SELECT \* FROM имя\_объекта   
WHERE поле IN (значение\_1, …, значение\_n)*

Оператор BETWEEN задает диапазон значений.

*SELECT \* FROM имя\_объекта   
WHERE поле BETWEEN значение\_1 AND значение\_n*

Для выполнения лабораторных работ 1-7 обязательно использование СУБД PostgreSQL и допустимо использование любой IDE, обеспечивающей удобную работу с базами данных (DBeaver, DataGrib или другие).

**ПРАКТИЧЕСКОЕ ЗАДАНИЕ**

1. Подключиться к СУБД Postgres, cоздать новую базу данных my\_db, создать в ней новую схему my\_schema. Созданные БД и схема должны быть выбраны в качестве основных для дальнейшей работы.
2. С помощью команды CREATE создать 4 таблицы: prod, cust, sal, ord. Таблицы должны иметь следующую структуру:

**Таблица продуктов - prod**

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Название поля* | *Тип данных* | *Ограничения* |
| Номер продукта | ***pnum*** | int | PRIMARY KEY |
| Название продукта | ***name*** | varchar(20) | NOT NULL |
| Вес продукта | ***weight*** | int | NOT NULL |
| Город производства | ***city*** | varchar(20) | NOT NULL |

**Таблица покупателей - cust**

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Название поля* | *Тип данных* | *Ограничения* |
| Номер покупателя | ***сnum*** | int | PRIMARY KEY |
| Имя покупателя | ***name*** | varchar(20) | NOT NULL |
| Рейтинг покупателя | ***rating*** | int | NOT NULL |
| Город покупателя | ***city*** | varchar(20) | NOT NULL |

**Таблица продавцов - sal**

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Название поля* | *Тип данных* | *Ограничения* |
| Номер продавца | ***snum*** | int | PRIMARY KEY |
| Название продавца | ***name*** | varchar(20) | NOT NULL |
| Комиссионные продавца | ***comm*** | numeric(7,2) | NOT NULL |
| Город продавца | ***city*** | varchar(20) | NOT NULL |

**Таблица заказов - ord**

|  |  |  |  |
| --- | --- | --- | --- |
| *Атрибут* | *Название поля* | *Тип данных* | *Ограничения* |
| Номер заказа | ***onum*** | int | PRIMARY KEY |
| Номер продукта | ***pnum*** | int | NOT NULL  FOREIGN KEY |
| Номер покупателя | ***cnum*** | int | NOT NULL  FOREIGN KEY |
| Номер продавца | ***snum*** | int | NOT NULL  FOREIGN KEY |
| Количество продукта | ***amt*** | int | NOT NULL |

1. С помощью команды INSERT заполнить таблицы следующими данными.

**Таблица prod**

|  |  |  |  |
| --- | --- | --- | --- |
| ***Pnum*** | ***name*** | ***weight*** | ***city*** |
| 1001 | Monitor | 2000 | Obninsk |
| 1002 | Keyboard | 500 | Yekaterinburg |
| 1003 | Mouse | 100 | Novosibirsk |
| 1004 | Printer | 1500 | Saint Petersburg |
| 1005 | Hard drive | 300 | Moscow |
| 1006 | Speakers | 700 | Novosibirsk |

**Таблица cust**

|  |  |  |  |
| --- | --- | --- | --- |
| ***cnum*** | ***name*** | ***rating*** | ***city*** |
| 2001 | Ivanov | 100 | Perm |
| 2002 | Petrov | 100 | Moscow |
| 2003 | Vasiliev | 200 | Yekaterinburg |
| 2004 | Dmitriev | 200 | Krasnoyarsk |
| 2005 | Skvortcov | 300 | Moscow |
| 2006 | Avdeev | 400 | Novosibirsk |
| 2007 | Smirnov | 100 | Omsk |

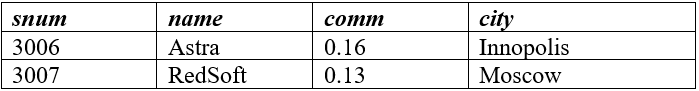
**Таблица sal**

|  |  |  |  |
| --- | --- | --- | --- |
| ***snum*** | ***name*** | ***comm*** | ***city*** |
| 3001 | DNS | 0.11 | Novosibirsk |
| 3002 | Citylink | 0.12 | Saint Petersburg |
| 3003 | MVideo | 0.15 | Yekaterinburg |
| 3004 | Inline | 0.13 | Vladivostok |
| 3005 | Elbrus | 0.11 | Moscow |
| 3006 | Astra | 0.16 | Innopolis |

**Таблица ord**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***onum*** | ***pnum*** | ***cnum*** | ***snum*** | ***amt*** | ***ord\_date*** |
| 4001 | 1001 | 2001 | 3001 | 2 | 2025-09-01 |
| 4002 | 1001 | 2002 | 3001 | 1 | 2025-09-01 |
| 4003 | 1001 | 2006 | 3002 | 10 | 2025-09-01 |
| 4004 | 1001 | 2003 | 3003 | 5 | 2025-09-01 |
| 4005 | 1001 | 2004 | 3004 | 5 | 2025-09-01 |
| 4006 | 1002 | 2001 | 3005 | 7 | 2024-12-31 |
| 4007 | 1002 | 2002 | 3001 | 8 | 2024-12-31 |
| 4008 | 1003 | 2001 | 3002 | 3 | 2025-09-01 |
| 4009 | 1003 | 2006 | 3003 | 1 | 2025-09-01 |
| 4010 | 1003 | 2007 | 3004 | 9 | 2025-09-01 |
| 4011 | 1003 | 2004 | 3003 | 6 | 2025-09-01 |
| 4012 | 1004 | 2002 | 3001 | 6 | 2025-09-01 |
| 4013 | 1004 | 2001 | 3002 | 4 | 2025-09-01 |
| 4014 | 1004 | 2001 | 3004 | 4 | 2025-09-01 |
| 4015 | 1004 | 2006 | 3003 | 3 | 2025-09-01 |
| 4016 | 1004 | 2004 | 3005 | 3 | 2025-09-01 |
| 4017 | 1004 | 2007 | 3002 | 2 | 2025-09-01 |
| 4018 | 1005 | 2001 | 3005 | 1 | 2025-09-01 |
| 4019 | 1006 | 2004 | 3005 | 2 | 2025-09-01 |
| 4020 | 1006 | 2003 | 3002 | 3 | 2025-09-01 |

1. Проверить корректность выполненных операций с помощью простейших выборок для каждой таблицы с помощью команды SELECT
2. С помощью команды INSERT добавить в таблицу sal две новых записи о продавцах с номерами 3006 и 3007 со следующими данными:



1. С помощью команды DELETE удалить из таблицы sal запись о продавце с номером 3007.
2. С помощью команды ALTER добавить в таблицу ord новый столбец даты заказа с именем ord\_date и типом данных date.
3. C помощью команды UPDATE установить дату всех заказов равной 1 сентября 2025 года.
4. C помощью команды UPDATE установить дату всех заказов для продукта с номером 1002 равной 31 декабря 2024 года.
5. Выполнить запросы к базе данных по выборке данных, согласно варианту бригады.

**ВАРИАНТЫ ЗАДАНИЙ ПО ВЫБОРКЕ ДАННЫХ**

*Вариант 1*

1. Вывести все строки из таблицы Заказов, для которых номер продукта равен 1001.
2. Вывести записи о продавцах с комиссионными ниже 0.14 со столбцами в следующем порядке: city, name, snum, comm.
3. Вывести без повторений номера всех продавцов, которые продавали товары покупателям с номерами больше 2004.
4. Вывести данные о всех покупателях с рейтингом меньше или равным 200, если они не находятся в Москве.
5. Вывести тремя различными способами все заказы продуктов с номерами 1002, 1003, 1004.

*Вариант 2*

1. Вывести все строки из таблицы Заказов, для которых номер продавца равен 3002.
2. Вывести записи о продавцах с комиссионными выше 0.11 со столбцами в следующем порядке: snum, city, name, comm.
3. Вывести без повторений номера всех покупателей, которые покупали товары у продавцов с номерами не менее 3003.
4. Вывести данные о всех продуктах с весов меньше или равным 1000, если они не производились в Москве.
5. Вывести тремя различными способами все заказы от продавцов с номерами 3003, 3004, 3005

*Вариант 3*

1. Вывести все строки из таблицы Заказов, для которых номер покупателя равен 2002.
2. Вывести записи о покупателях с рейтингом выше или равным 200 со столбцами в следующем порядке: сnum, city, name, rating
3. Вывести без повторений номера всех продуктов, которые заказывали покупатели с номерами не более 2005.
4. Вывести данные о всех покупателях с рейтингом меньше 200, если они находятся в Санкт-Петербурге или Новосибирске.
5. Вывести тремя различными способами все заказы покупателей с номерами 2004, 2005, 2006.

*Вариант 4*

1. Вывести все строки из таблицы Заказов, для которых номер продукта не равен 1005.
2. Вывести записи о покупателях с рейтингом ниже или равным 150 со столбцами в следующем порядке: rating, name, сnum, city.
3. Вывести без повторений номера всех продуктов, которые поставляли продавцы с номерами более 3003.
4. Вывести данные о всех продуктах с весов более 100, если они не производились в Санкт-Петербурге.
5. Вывести тремя различными способами все заказы от продавцов с номерами 3002, 3003, 3004.

*Вариант 5*

1. Вывести все строки из таблицы Заказов, для которых номер продавца не равен 3005.
2. Вывести записи о продуктах с весом строго больше 500 со столбцами в следующем порядке: city, name, pnum, weight.
3. Вывести без повторений номера всех продавцов, которые поставляли продукты с номерами не более 1001.
4. Вывести данные о всех продавцах с комиссионными менее или равным 0.13, если они не находятся в Екатеринбурге.
5. Вывести тремя различными способами все заказы покупателей с номерами 2005, 2005, 2007.

*Вариант 6*

1. Вывести все строки из таблицы Заказов, для которых номер покупателя не равен 2003.
2. Вывести записи о продуктах с весом меньше или равным 700 со столбцами в следующем порядке: pnum, weight, name, city.
3. Вывести без повторений номера всех покупателей, которые заказывали продукты с номерами более или равным 1006.
4. Вывести данные о всех продавцах с комиссионными более 0.12, если они находятся во Владивостоке или Новосибирске.
5. Вывести тремя различными способами все заказы продуктов с номерами 1003, 1004, 1005.