# Основы устр. БД

## Модели данных

1. Модель данных - логическая структура, в соответствии с которой данные в БД хранятся.

Примеры моделей данных:

* иерархическая
* сетевая
* реляционная
* постреляционная
* многомерная
* Объектно - ориентированная

## Реляционная модель данных

### Связываение таблиц

#### 1:n

# Oracle

## Контейнер

### Поднятие контейнера

Note:

несмотря на то какой порт указать при создании контейнера(это другое) при подключении к нему контейнер слушает именно 1521 по умолчанию

#### Free lite

docker pull container-registry.oracle.com/database/free:latest-lite

docker exec -it b79cf7fdf92271b8606e5a51712c33baa71877f81c2e75d2d91c2c9629136928 bash

Почему то в lite версии можно указать имя БД и потом всегда по имени обращаться к ней.

#### Free

docker pull container-registry.oracle.com/database/free:latest

docker exec -it b79cf7fdf92271b8606e5a51712c33baa71877f81c2e75d2d91c2c9629136928 bash

А в обычной free версии контейнера нельзя при создании указать имя БД и требуется смотреть имена сервисов контейнера - lsnrctl status внутри контейнера:

docker exec -it 57b9369e48566de0ca509d50dbb532428d822513415a24c51d1200834df385ef bash

bash-4.4$ lsnrctl status

Но по умолчанию обычная версия имеет имя БД: FREE. И можно через FREE везде обращаться.

### !Подключение к БД из контейнера

Connecting from Within the Container

You can connect to the Oracle Database by running a SQL\*Plus command from within the container using one of the following commands:

podman exec -it <oracle-db> sqlplus sys/<your\_password>@FREE as sysdba

podman exec -it <oracle-db> sqlplus system/<your\_password>@FREE

podman exec -it <oracle-db> sqlplus pdbadmin/<your\_password>@FREEPDB1

## Роли пользователей в БД

### Создание нового пользователя

Шаг 1: Подключись к Oracle как администратор

sqlplus system/YourSystemPassword@localhost:1521/FREEPDB1

Шаг 2: Создание пользователя

-- Создаём пользователя

CREATE USER excel\_user IDENTIFIED BY Excel123;

-- Даём базовые права: вход и создание таблиц

GRANT CONNECT, RESOURCE TO excel\_user;

-- если табличное пространство USERS существует

ALTER USER excel\_user QUOTA UNLIMITED ON USERS;

Шаг 3: Проверить пространства и выдать свободное пользователю

1) Проверка

Для создания места в табличном пространстве для данного пользователя нужно проверить из администратора:

Доступные пространства

SELECT tablespace\_name FROM dba\_tablespaces;

Пример вывода:

TABLESPACE\_NAME

------------------------------

SYSTEM

SYSAUX

UNDOTBS1

TEMP

По умолчанию все эти пространства нельзя использовать.

Если нет ДРУГИХ пространств для пользователей, то нужно создать его и присвоить какому либо пользователю. Для этого:

Посмотри, где хранятся файлы базы данных

SELECT name FROM v$datafile;

/opt/oracle/oradata/FREE/FREEPDB1/system01.dbf

/opt/oracle/oradata/FREE/FREEPDB1/sysaux01.dbf

/opt/oracle/oradata/FREE/FREEPDB1/undotbs01.dbf

выбрать один из путей и создать новое пространство

CREATE TABLESPACE user\_data

DATAFILE '/opt/oracle/oradata/FREE/FREEPDB1/user\_data01.dbf'

SIZE 50M AUTOEXTEND ON NEXT 10M MAXSIZE 1G; # указали размер пространства

2) Выдать это пространство пользователю

ALTER USER excel\_user QUOTA UNLIMITED ON user\_data;

Шаг 4: Новый пользователь создает таблицу (в выданном ему пространстве - всё настроено автоматически для пользователя)

sqlplus excel\_user/Excel123@localhost:1521/FREEPDB1

CREATE TABLE test\_excel (

id NUMBER PRIMARY KEY,

name VARCHAR2(100),

created\_at DATE DEFAULT SYSDATE

);

## Подключение к БД

### Из контейнера

Connecting from Within the Container

You can connect to the Oracle Database by running a SQL\*Plus command from within the container using one of the following commands:

podman exec -it <oracle-db> sqlplus sys/<your\_password>@FREE as sysdba

podman exec -it <oracle-db> sqlplus system/<your\_password>@FREE

podman exec -it <oracle-db> sqlplus pdbadmin/<your\_password>@FREEPDB1

### Python

pip install oracledb

import oracledb

# Thick подключение

# oracledb.init\_oracle\_client(lib\_dir="D:\programs\oracle\Oracle Instance Client

# (OIC)\instantclient-basic-windows.x64-23.8.0.25.04\instantclient\_23\_8")

conn = oracledb.connect(

user="system",

password="MyPass123",

dsn="localhost:1521/oracle"

)

print("Успешное подключение!!!")

cur = conn.cursor()

cur.execute("SELECT \* FROM test\_ole")

for row in cur:

print(row)

cur.close()

conn.close()

## Создание таблиц

нужно войти в БД под конкретным пользователем, после чего можно создавать таблицы:

CREATE TABLE test\_ole (

id NUMBER PRIMARY KEY,

name VARCHAR2(100),

created\_at DATE

);

INSERT INTO test\_ole VALUES (1, 'Alice', SYSDATE);

INSERT INTO test\_ole VALUES (2, 'Bob', SYSDATE);

INSERT INTO test\_ole VALUES (3, 'Charlie', SYSDATE);

### system создает таблицы для пользователей

Как system пользователь ты можешь сделать так:

sql

Копировать код

CREATE TABLE users\_table (

id NUMBER PRIMARY KEY,

name VARCHAR2(100)

) TABLESPACE USERS;

Если у SYSTEM есть права на табличное пространство (а у него они по умолчанию есть), команда выполнится.

⚠️ Но почему не стоит так делать?

SYSTEM — это административный пользователь, предназначенный для управления БД, а не для хранения пользовательских данных.

Хранение пользовательских объектов (таблиц, индексов и т.п.) под пользователем SYSTEM нарушает best practices Oracle.

Это может привести к загрязнению системной схемы, усложнить бэкапы, миграции и поддержку.