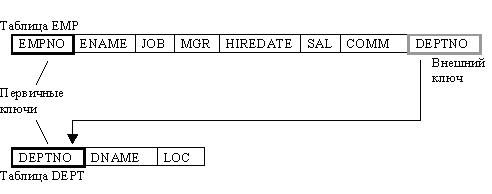
**Лабораторная работа № 2. Загрузка учебных схем (EMP, HR).**

**Задание 1.**

|  |  |
| --- | --- |
| * вести файл протокола spool. txt   SQL> set echo on  SQL> spool z:\BD\spool.txt | * завершить протокол лаб. работы   SQL> spool off |
| * создать пользователя   SQL> create user c##user1 identified by user1  default tablespace system  quota unlimited on system;   * изменить пароль   SQL> alter user c##user1 identified by test;   * разблокировать   SQL> alter user c##user1 account unlock; | - просмотреть пользователей  SQL>select username from sys.dba\_users;  -предоставить роль DBA  SQL> grant dba to user1;  -уничтожить пользователя  SQL> drop user c##user1 cascade;  -показать пользователя  SQL>show user |
| - просмотреть названия таблиц в схеме USER1  SQL>select table\_name from sys.all\_tables where owner ='C##USER1'; | - просмотреть из system структуру таблицы regions схемы HR  SQL>desc HR.regions  - просмотреть из system содержимое таблицы regions схемы HR  SQL>select \* from HR.regions |

**Задание 2.** Сгенерировать таблицы EMP в SQL Plus Command line



--

-- Copyright (c) Oracle Corporation 1988, 2000. All Rights Reserved.

--

-- NAME

-- demobld.sql

--

-- DESCRIPTION

-- This script creates the SQL\*Plus demonstration tables in the

-- current schema. It should be STARTed by each user wishing to

-- access the tables. To remove the tables use the demodrop.sql

-- script.

--

-- USAGE

-- From within SQL\*Plus, enter:

-- START demobld.sql

SET TERMOUT ON

PROMPT Building demonstration TABLES. Please wait.

SET TERMOUT OFF

DROP TABLE EMP;

DROP TABLE DEPT;

DROP TABLE DEPT1;

DROP TABLE BONUS;

DROP TABLE SALGRADE;

DROP TABLE DUMMY;

CREATE TABLE EMP

(EMPNO NUMBER(4) NOT NULL,

ENAME VARCHAR2(10),

JOB VARCHAR2(9),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7, 2),

COMM NUMBER(7, 2),

DEPTNO NUMBER(2));

INSERT INTO EMP VALUES

(7369, 'SMITH', 'CLERK', 7902,

TO\_DATE('17-12-1980', 'DD-MM-YYYY'), 800, NULL, 20);

INSERT INTO EMP VALUES

(7369, 'SMITH', 'CLERK', 7902,

TO\_DATE('17-12-1980', 'DD-MM-YYYY'), 800, NULL, 20);

INSERT INTO EMP VALUES

(7499, 'ALLEN', 'SALESMAN', 7698,

TO\_DATE('20-02-1981', 'DD-MM-YYYY'), 1600, 300, 30);

INSERT INTO EMP VALUES

(7521, 'WARD', 'SALESMAN', 7698,

TO\_DATE('22-02-1981', 'DD-MM-YYYY'), 1250, 500, 30);

INSERT INTO EMP VALUES

(7566, 'JONES', 'MANAGER', 7839,

TO\_DATE('2-4-1981', 'DD-MM-YYYY'), 2975, NULL, 20);

INSERT INTO EMP VALUES

(7654, 'MARTIN', 'SALESMAN', 7698,

TO\_DATE('28-9-1981', 'DD-MM-YYYY'), 1250, 1400, 30);

INSERT INTO EMP VALUES

(7698, 'BLAKE', 'MANAGER', 7839,

TO\_DATE('1-5-1981', 'DD-MM-YYYY'), 2850, NULL, 30);

INSERT INTO EMP VALUES

(7782, 'CLARK', 'MANAGER', 7839,

TO\_DATE('9-6-1981', 'DD-MM-YYYY'), 2450, NULL, 10);

INSERT INTO EMP VALUES

(7788, 'SCOTT', 'ANALYST', 7566,

TO\_DATE('09-12-1982', 'DD-MM-YYYY'), 3000, NULL, 20);

INSERT INTO EMP VALUES

(7839, 'KING', 'PRESIDENT', NULL,

TO\_DATE('17-11-1981', 'DD-MM-YYYY'), 5000, NULL, 10);

INSERT INTO EMP VALUES

(7844, 'TURNER', 'SALESMAN', 7698,

TO\_DATE('8-9-1981', 'DD-MM-YYYY'), 1500, 0, 30);

INSERT INTO EMP VALUES

(7876, 'ADAMS', 'CLERK', 7788,

TO\_DATE('12-1-1983', 'DD-MM-YYYY'), 1100, NULL, 20);

INSERT INTO EMP VALUES

(7900, 'JAMES', 'CLERK', 7698,

TO\_DATE('3-12-1981', 'DD-MM-YYYY'), 950, NULL, 30);

INSERT INTO EMP VALUES

(7902, 'FORD', 'ANALYST', 7566,

TO\_DATE('3-12-1981', 'DD-MM-YYYY'), 3000, NULL, 20);

INSERT INTO EMP VALUES

(7934, 'MILLER', 'CLERK', 7782,

TO\_DATE('23-1-1982', 'DD-MM-YYYY'), 1300, NULL, 10);

INSERT INTO EMP VALUES

(7934, 'MILLER', 'CLERK', 7782,

TO\_DATE('23-1-1982', 'DD-MM-YYYY'), 1300, NULL, 10);

CREATE TABLE DEPT

(DEPTNO NUMBER(2),

DNAME VARCHAR2(14),

LOC VARCHAR2(13) );

INSERT INTO DEPT VALUES (10, 'ACCOUNTING', 'NEW YORK');

INSERT INTO DEPT VALUES (20, 'RESEARCH', 'DALLAS');

INSERT INTO DEPT VALUES (30, 'SALES', 'CHICAGO');

INSERT INTO DEPT VALUES (40, 'OPERATIONS', 'BOSTON');

CREATE TABLE BONUS

(ENAME VARCHAR2(10),

JOB VARCHAR2(9),

SAL NUMBER,

COMM NUMBER);

CREATE TABLE SALGRADE

(GRADE NUMBER,

LOSAL NUMBER,

HISAL NUMBER);

INSERT INTO SALGRADE VALUES (1, 700, 1200);

INSERT INTO SALGRADE VALUES (2, 1201, 1400);

INSERT INTO SALGRADE VALUES (3, 1401, 2000);

INSERT INTO SALGRADE VALUES (4, 2001, 3000);

INSERT INTO SALGRADE VALUES (5, 3001, 9999);

CREATE TABLE DUMMY

(DUMMY NUMBER);

INSERT INTO DUMMY VALUES (0);

COMMIT;

SET TERMOUT ON

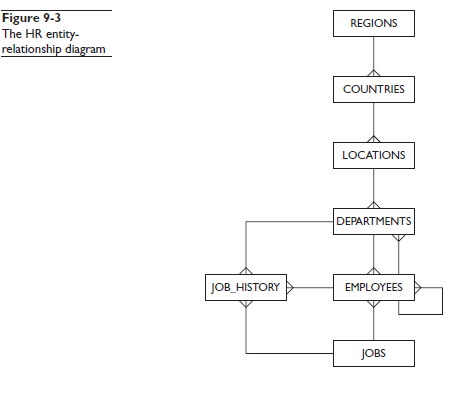
PROMPT DEMONSRTRATION TABLE BUILD IS COMPLETE

**Задание 4. Посмотреть сгенерированные таблицы в sqldeveloper (диск E:), ER диаграмму (model)**

**Задание 5.**

Схема HR состоит из семи таблиц, связанных первичными и внешними ключами. На рисунке 9-3 отображены отношения между таблицами как на диаграмме сущность-связь.

Два из отношений на рисунке 9-3 могут быть сразу не совсем понятны. Первое, это отношение один-ко-многим таблицы EMPLOYEE к таблице EMPLOYEE. Это self-referencing внешний ключ. Это значит что много сотрудников может быть связано с одним сотрудником, и это основано на том факте что у многих сотрудников может быть один менеджер, но менеджер это также сотрудник. Эта связь реализована за счёт столбца внешнего ключа manager\_id и первичного ключа employee\_id. Вторая связь которая возможно требует дополнительного пояснения это двунаправленная  связь между DEPARTMENTS  и EMPLOYEES. Многие сотрудники могут работать в одном департаменте, и один сотрудник может быть менеджером многих департаментов.



В таблице 9-1 отображены столбцы таблиц схемы HR, используя описанную ранее нотацию для отображения первичных ключей (#), внешних ключей (\), обязательности (\*) или необязательности (o) значения.

Таблицы в схеме

REGIONS содержит строки для георафических областей

COUNTRIES содержит строки для каждой страны, которые могут закрепляться за регионом

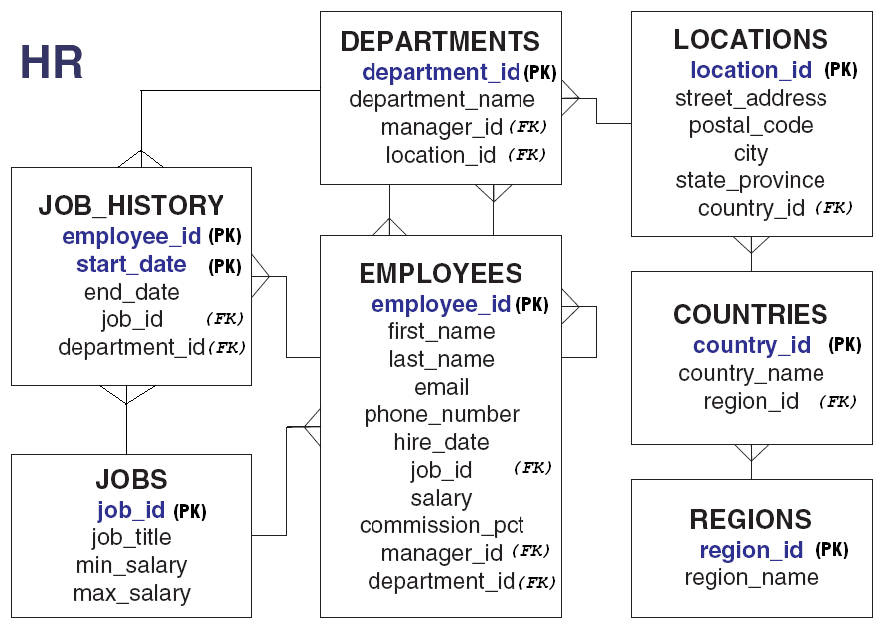
LOCATIONS содержат адреса, которые могут принадлежать стране

DEPARTAMENTS хранит информацию об отделах, с необязательным адресом и необязательным полем менеджера (менеджер должен существовать как сотрудник)

EMPLOYEES хранит строки для каждого сотрудника, каждому из которых должна быть назначена должность и необязательно менеджер и департамент. Менеджер в свою очередь должен быть в таблице сотрудников

JOBS содержит информацию о должностях в компании. Много сотрудников могут иметь одинаковую должность

JOB\_HISTORY хранит информацию о предыдущих должностях сотрудника, уникально определяемую полями employee\_id и start\_date; сотруднику не может быть назначено несколько должностей в один момент времени. Каждая запись в истории ссылается на сотрудника с одной должностью в определённый момент и возможной принадлежности какому-либо отделу. Мы будем использовать HR схему для примеров и упражнений, так что она должна быть создана.



Задание – создать три связанные таблицы.

|  |  |  |
| --- | --- | --- |
|  | | |
| CREATE TABLE regions  ( region\_id NUMBER  CONSTRAINT region\_id\_nn NOT NULL  , region\_name VARCHAR2(25)  );  CREATE UNIQUE INDEX reg\_id\_pk  ON regions (region\_id);  ALTER TABLE regions  ADD ( CONSTRAINT reg\_id\_pk  PRIMARY KEY (region\_id)  ) ; | CREATE TABLE countries  ( country\_id CHAR(2)  CONSTRAINT country\_id\_nn NOT NULL  , country\_name VARCHAR2(40)  , region\_id NUMBER  , CONSTRAINT country\_c\_id\_pk  PRIMARY KEY (country\_id)  )  ORGANIZATION INDEX;  ALTER TABLE countries  ADD ( CONSTRAINT countr\_reg\_fk  FOREIGN KEY (region\_id)  REFERENCES regions(region\_id)  ) ; | CREATE TABLE locations  ( location\_id NUMBER(4)  , street\_address VARCHAR2(40)  , postal\_code VARCHAR2(12)  , city VARCHAR2(30)  CONSTRAINT loc\_city\_nn NOT NULL  , state\_province VARCHAR2(25)  , country\_id CHAR(2)  ) ;  CREATE UNIQUE INDEX loc\_id\_pk  ON locations (location\_id) ;  ALTER TABLE locations  ADD ( CONSTRAINT loc\_id\_pk  PRIMARY KEY (location\_id)  , CONSTRAINT loc\_c\_id\_fk  FOREIGN KEY (country\_id)  REFERENCES countries(country\_id) ) ; |
| REM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*insert data into the REGIONS table  Prompt \*\*\*\*\*\* Populating REGIONS table ....  INSERT INTO regions VALUES  ( 1  , 'Europe'  );  INSERT INTO regions VALUES  ( 2  , 'Americas'  );  INSERT INTO regions VALUES  ( 3  , 'Asia'  );  INSERT INTO regions VALUES  ( 4  , 'Middle East and Africa'  ); | REM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*insert data into the COUNTRIES table  Prompt \*\*\*\*\*\* Populating COUNTIRES table ....  INSERT INTO countries VALUES  ( 'IT'  , 'Italy'  , 1  );  INSERT INTO countries VALUES  ( 'JP'  , 'Japan'  , 3  );  INSERT INTO countries VALUES  ( 'US'  , 'United States of America'  , 2  );  INSERT INTO countries VALUES  ( 'CA'  , 'Canada'  , 2  );  INSERT INTO countries VALUES  ( 'CN'  , 'China'  , 3  );  INSERT INTO countries VALUES  ( 'IN'  , 'India'  , 3  );  INSERT INTO countries VALUES  ( 'AU'  , 'Australia'  , 3  );  INSERT INTO countries VALUES  ( 'ZW'  , 'Zimbabwe'  , 4  );  INSERT INTO countries VALUES  ( 'SG'  , 'Singapore'  , 3  );  INSERT INTO countries VALUES  ( 'UK'  , 'United Kingdom'  , 1  );  INSERT INTO countries VALUES  ( 'FR'  , 'France'  , 1  );  INSERT INTO countries VALUES  ( 'DE'  , 'Germany'  , 1  );  INSERT INTO countries VALUES  ( 'ZM'  , 'Zambia'  , 4  );  INSERT INTO countries VALUES  ( 'EG'  , 'Egypt'  , 4  );  INSERT INTO countries VALUES  ( 'BR'  , 'Brazil'  , 2  );  INSERT INTO countries VALUES  ( 'CH'  , 'Switzerland'  , 1  );  INSERT INTO countries VALUES  ( 'NL'  , 'Netherlands'  , 1  );  INSERT INTO countries VALUES  ( 'MX'  , 'Mexico'  , 2  );  INSERT INTO countries VALUES  ( 'KW'  , 'Kuwait'  , 4  );  INSERT INTO countries VALUES  ( 'IL'  , 'Israel'  , 4  );  INSERT INTO countries VALUES  ( 'DK'  , 'Denmark'  , 1  );  INSERT INTO countries VALUES  ( 'HK'  , 'HongKong'  , 3  );  INSERT INTO countries VALUES  ( 'NG'  , 'Nigeria'  , 4  );  INSERT INTO countries VALUES  ( 'AR'  , 'Argentina'  , 2  );  INSERT INTO countries VALUES  ( 'BE'  , 'Belgium'  , 1  ); | REM \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*insert data into the LOCATIONS table  Prompt \*\*\*\*\*\* Populating LOCATIONS table ....  INSERT INTO locations VALUES  ( 1000  , '1297 Via Cola di Rie'  , '00989'  , 'Roma'  , NULL  , 'IT'  );  INSERT INTO locations VALUES  ( 1100  , '93091 Calle della Testa'  , '10934'  , 'Venice'  , NULL  , 'IT'  );  INSERT INTO locations VALUES  ( 1200  , '2017 Shinjuku-ku'  , '1689'  , 'Tokyo'  , 'Tokyo Prefecture'  , 'JP'  );  INSERT INTO locations VALUES  ( 1300  , '9450 Kamiya-cho'  , '6823'  , 'Hiroshima'  , NULL  , 'JP'  );  INSERT INTO locations VALUES  ( 1400  , '2014 Jabberwocky Rd'  , '26192'  , 'Southlake'  , 'Texas'  , 'US'  );  INSERT INTO locations VALUES  ( 1500  , '2011 Interiors Blvd'  , '99236'  , 'South San Francisco'  , 'California'  , 'US'  );  INSERT INTO locations VALUES  ( 1600  , '2007 Zagora St'  , '50090'  , 'South Brunswick'  , 'New Jersey'  , 'US'  );  INSERT INTO locations VALUES  ( 1700  , '2004 Charade Rd'  , '98199'  , 'Seattle'  , 'Washington'  , 'US'  );  INSERT INTO locations VALUES  ( 1800  , '147 Spadina Ave'  , 'M5V 2L7'  , 'Toronto'  , 'Ontario'  , 'CA'  );  INSERT INTO locations VALUES  ( 1900  , '6092 Boxwood St'  , 'YSW 9T2'  , 'Whitehorse'  , 'Yukon'  , 'CA'  );  INSERT INTO locations VALUES  ( 2000  , '40-5-12 Laogianggen'  , '190518'  , 'Beijing'  , NULL  , 'CN'  );  INSERT INTO locations VALUES  ( 2100  , '1298 Vileparle (E)'  , '490231'  , 'Bombay'  , 'Maharashtra'  , 'IN'  );  INSERT INTO locations VALUES  ( 2200  , '12-98 Victoria Street'  , '2901'  , 'Sydney'  , 'New South Wales'  , 'AU'  );  INSERT INTO locations VALUES  ( 2300  , '198 Clementi North'  , '540198'  , 'Singapore'  , NULL  , 'SG'  );  INSERT INTO locations VALUES  ( 2400  , '8204 Arthur St'  , NULL  , 'London'  , NULL  , 'UK'  );  INSERT INTO locations VALUES  ( 2500  , 'Magdalen Centre, The Oxford Science Park'  , 'OX9 9ZB'  , 'Oxford'  , 'Oxford'  , 'UK'  );  INSERT INTO locations VALUES  ( 2600  , '9702 Chester Road'  , '09629850293'  , 'Stretford'  , 'Manchester'  , 'UK'  );  INSERT INTO locations VALUES  ( 2700  , 'Schwanthalerstr. 7031'  , '80925'  , 'Munich'  , 'Bavaria'  , 'DE'  );  INSERT INTO locations VALUES  ( 2800  , 'Rua Frei Caneca 1360 '  , '01307-002'  , 'Sao Paulo'  , 'Sao Paulo'  , 'BR'  );  INSERT INTO locations VALUES  ( 2900  , '20 Rue des Corps-Saints'  , '1730'  , 'Geneva'  , 'Geneve'  , 'CH'  );  INSERT INTO locations VALUES  ( 3000  , 'Murtenstrasse 921'  , '3095'  , 'Bern'  , 'BE'  , 'CH'  );  INSERT INTO locations VALUES  ( 3100  , 'Pieter Breughelstraat 837'  , '3029SK'  , 'Utrecht'  , 'Utrecht'  , 'NL'  );  INSERT INTO locations VALUES  ( 3200  , 'Mariano Escobedo 9991'  , '11932'  , 'Mexico City'  , 'Distrito Federal,'  , 'MX'  ); |

**Задание 6. Посмотреть созданные таблицы в sqldeveloper (диск E:), ER диаграмму (model)**

**Задание 7. ЗАГРУЗИТЬ УЧЕБНУЮ БАЗУ**

SQL> set echo on

SQL> spool z:\BD\spool. Txt

DROP USER c##hr CASCADE;

CREATE USER c##hr IDENTIFIED BY HR;

ALTER USER c##hr DEFAULT TABLESPACE USERS QUOTA 100M ON USERS;

ALTER USER c##hr TEMPORARY TABLESPACE TEMP QUOTA UNLIMITED ON USERS;

GRANT CONNECT TO c##hr;

GRANT RESOURCE TO c##hr;

GRANT create any table TO c##hr;

GRANT create any view TO c##hr;

CONNECT sys/orapass AS SYSDBA;

GRANT execute ON sys.dbms\_stats TO c##hr;

CONNECT c##hr/&pass

ALTER SESSION SET NLS\_LANGUAGE=American;

ALTER SESSION SET NLS\_TERRITORY=America;

--

-- create tables, sequences and constraint

**Задание 8. Посмотреть созданные таблицы в sqldeveloper (диск E:), ER диаграмму (model).**