1 数据库范式

第一范式1NF：表的每一列都是不可分割的基本数据项，同一列中不能有多个值；
第二范式2NF：必须首先满足第一范式，要求实体的属性完全依赖于主关键字；
第三范式3NF：必须首先满足第二范式，第三范式要求一个数据库表中不包含已在其它表中已包含的非主关键字信息。

学生

学号	姓名	性别	出生日期	家庭电话	个人电话	课程号	课程名称

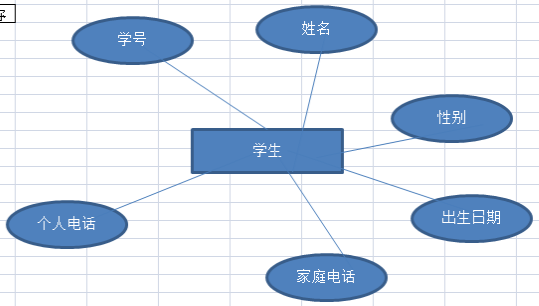
课程

课程号	课程名称	开课日期

学生-课程

ID	学号	课程号
	1	C1
	2	C1
	1	C2
	2	C2

PD
Erwin
VISIO



CDM
LDM
PDM

实体/表之间的对应关系：

- 一对一
- 一对多
- 多对多

2 结构化查询语言（SQL）

```
select ....
   from ....
 where ....
group by ...
having ....
 order by ....
```

```
SELECT * FROM scott.emp;
```

```
SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno FROM scott.emp;
```

```
SELECT empno, ename, sal FROM scott.emp;
```

```
SELECT * FROM scott.emp t;

SELECT t.ename, t.empno, t.sal FROM scott.emp t;

SELECT t.ename AS 姓名, t.empno 工号, t.sal 工资 FROM scott.emp t;

SELECT t.ename "EMP NAME", t.empno "EmpNo", t.sal "Sal" FROM scott.emp t;

SELECT t.ename, t.empno, t.sal FROM scott.emp t WHERE t.sal > 3000;

--且 and
SELECT t.ename, t.empno, t.sal, t.deptno
  FROM scott.emp t
 WHERE t.sal >= 1250
   AND t.sal <= 3000
   AND t.deptno = 30;

--或 or

SELECT t.ename, t.empno, t.sal, t.deptno
  FROM scott.emp t
 WHERE t.deptno = 20
    OR sal > 2000;

--非

SELECT t.ename, t.empno, t.sal, t.deptno
  FROM scott.emp t
 WHERE --t.deptno !=20
 t.deptno <> 20;

/*
  该SQL功能是查询员工
  工资在1250到3000的信息
```

```
*/
SELECT t.ename, t.empno, t.sal
  FROM scott.emp t
 WHERE t.sal BETWEEN 1250 AND 3000;

/*
=
>
<
>=
<= != <>
BETWEEN xxx AND xxx;
*/

SELECT t.ename, t.empno, t.sal, t.deptno
  FROM scott.emp t
 WHERE t.deptno = 30
 ORDER BY sal;

SELECT t.ename, t.empno, t.sal, t.deptno
  FROM scott.emp t
 WHERE t.deptno = 30
 ORDER BY sal DESC;

SELECT t.ename, t.empno, t.sal, t.deptno, t.comm
  FROM scott.emp t
 WHERE t.deptno = 30
 ORDER BY comm DESC NULLS LAST;

SELECT t.ename, t.empno, t.sal, t.deptno, t.comm "奖金"
  FROM scott.emp t
 WHERE t.deptno = 30
 ORDER BY "奖金";

SELECT t.ename, t.empno, t.sal, t.deptno, t.comm "奖金"
```

```
  FROM scott.emp t
 WHERE t.deptno = 30
 ORDER BY "奖金";

SELECT t.ename, t.empno, t.sal, t.deptno, t.comm "奖金"
  FROM scott.emp t
 WHERE t.deptno = 30
 ORDER BY 5;

SELECT t.ename, t.empno, t.sal, t.deptno, t.comm
  FROM scott.emp t
 ORDER BY deptno, sal DESC;

SELECT t.ename, t.empno, t.sal, t.deptno, t.comm
  FROM scott.emp t
 ORDER BY sal DESC, deptno;

SELECT t.deptno FROM scott.emp t GROUP BY t.deptno ORDER BY 1;

SELECT DISTINCT t.deptno FROM scott.emp t;

SELECT DISTINCT t.empno, t.deptno FROM scott.emp t;

SELECT t.sal + 1000, t.* FROM scott.emp t;

SELECT sal, t.sal + 1000 sal1, t.* FROM scott.emp t;

--SMITH CLERK  ||

SELECT t.ename || ' - ' || t.job, t.* FROM scott.emp t;

SELECT * FROM scott.emp t WHERE t.ename LIKE 'W%';

SELECT * FROM scott.emp t WHERE t.ename LIKE '%N';
```

```
SELECT rownum, t.* FROM scott.emp t;

SELECT rownum, m.* FROM (SELECT t.* FROM scott.emp t ORDER BY sal) m;

SELECT ROWID, t.* FROM scott.emp t;

--AAAWwTAAGAAAADDAAA

SELECT * FROM dba_data_files t WHERE t.relative_fno = 6;

SELECT dbms_rowid.rowid_relative_fno(row_id => 'AAAWwTAAGAAAADDAAA'),
       dbms_rowid.rowid_row_number(row_id => 'AAAWwTAAGAAAADDAAA'),
       dbms_rowid.rowid_block_number(row_id => 'AAAWwTAAGAAAADDAAA')
  FROM dual;
```

3 数据操纵语言（DML)

```
UPDATE
   SET 1 = xxxx, 2 = xxx .. ..
 WHERE ;

DELETE
 WHERE  ;

INSERT INTO (12...)
     VALUES ();

INSERT INTO (12...)
     SELECT 1,2.....
       FROM ()
      WHERE ;

INSERT INTO (12...)
SELECT 1,2..... from dual;
```

```
SELECT * FROM scott.emp t
--update scott.emp t  set sal=sal*1.1
WHERE t.empno = 7369;

SELECT * FROM scott.emp t
--update scott.emp t  set sal=sal*1.1,comm=2000
WHERE t.empno = 7369;

INSERT INTO scott.emp(empno,ename,job,hiredate,sal,deptno)
VALUES(7935,'Alen','IT',SYSDATE,1000,10);

INSERT INTO scott.emp(empno,ename,job,mgr,hiredate,sal,comm,deptno)
SELECT 7936, 'Tom', job, mgr, hiredate, sal, comm, deptno
  FROM scott.emp
 WHERE empno = 7369;

SELECT * FROM scott.emp
--delete scott.emp
WHERE empno = 7936;

SELECT * FROM scott.emp
--delete scott.emp
WHERE empno = 7935;
COMMIT;
```

4 数据定义语言（DDL)

4.1 表

```
-- Create table
create table TEST
(
  id          number,
```

```
  name        varchar2(100),
  create_date date default sysdate,
  creator     varchar2(100) default 'SYS'
);
-- Add comments to the table
comment on table TEST
  is '这是测试表';
-- Add comments to the columns
comment on column TEST.id
  is 'ID';
comment on column TEST.name
  is '姓名';
comment on column TEST.create_date
  is '创建时间';
comment on column TEST.creator
  is '创建人';

--CTAS

CREATE TABLE emp_test AS
SELECT * from scott.emp WHERE 1=2;

CREATE TABLE emp_test1 AS
SELECT * from scott.emp ;

SELECT * from emp_test1;

SELECT dbms_metadata.get_ddl(object_type => 'TABLE',
                             NAME        => 'EMP_TEST1',
                             SCHEMA      => USER)
  FROM dual;

--修改表 增加列
alter table EMP_TEST1 add create_date date;
alter table EMP_TEST1 add creator varchar2(100);
```

```
-- 修改表 删除列

ALTER TABLE EMP_TEST1 DROP COLUMN create_date;
ALTER TABLE EMP_TEST1 DROP COLUMN creator;
--OR
ALTER TABLE EMP_TEST1 DROP (create_date,creator);

-- 截断表
TRUNCATE TABLE EMP_TEST1;

--删除表

DROP TABLE EMP_TEST1;

SELECT * from User_Recyclebin;
FLASHBACK TABLE EMP_TEST1 TO BEFORE DROP;

SELECT * from EMP_TEST1;

DROP TABLE EMP_TEST1 PURGE;

--主键
alter table TEST
  add constraint pk_test_id primary key (ID);

--创建索引
CREATE INDEX idx_test_name ON TEST (NAME);
```

4.2视图

```
SQL> grant create view to scott;

--创建视图
CREATE VIEW emp_vw AS
```

```
SELECT empno,ename,job,mgr,deptno from emp;

SELECT * from emp_vw;

--删除视图
DROP VIEW emp_vw;
```

4.3 序列

```
--序列创建及使用
CREATE SEQUENCE seq_test;

-- Create sequence
create sequence SEQ_TEST
minvalue 1
maxvalue 9999999999999999999999999999
start with 1
increment by 1
cache 20;

SELECT SEQ_TEST.Nextval from dual;
SELECT SEQ_TEST.Currval from dual;

INSERT INTO TEST VALUES(SEQ_TEST.Nextval,'Alen',SYSDATE,'sys');
COMMIT;
SELECT * from TEST;

INSERT INTO TEST(ID,NAME) VALUES(SEQ_TEST.Nextval,'Alen');
COMMIT;

INSERT INTO TEST(ID,NAME) VALUES(SEQ_TEST.Nextval,'Tom');
COMMIT;

--删除序列
DROP SEQUENCE SEQ_TEST;
```

4.4 同义词

```
--sys用户下授权
SQL> grant select on oe.ORDERS to scott;

Grant succeeded.

SQL> grant CREATE SYNONYM to scott;

Grant succeeded.
--scott 用户下操作
SELECT * from dba_tables t
WHERE t.owner='OE';

SELECT * from oe.ORDERS;

--创建同义词
CREATE SYNONYM t_order FOR oe.ORDERS;

SELECT * from t_order;

--删除同义词
DROP SYNONYM ORDERS;
--查看同义词
SELECT * from User_Synonyms;
```

4.5 DBLINK

```
--SYS下授权
SQL> grant create database link to scott;

Grant succeeded.
--创建DBLink
```

```
create database link SCOTT_SCOTT
  connect to SCOTT
  using '192.168.56.102:1521/orcl';

SELECT * from oe.inventories@SCOTT_SCOTT;

CREATE SYNONYM inventories FOR oe.inventories@SCOTT_SCOTT;

SELECT * from inventories;

SELECT * from User_Synonyms;

--删除DBLink
DROP database link SCOTT_SCOTT;
```

4.6 Scheduler

```
--SYS下授权

--创建调度

begin
  sys.dbms_scheduler.create_job(job_name            => 'SCOTT.JOB_TEST',
                                job_type            => 'PLSQL_BLOCK',
                                job_action          => 'BEGIN
      INSERT INTO test
        (id, NAME)
      VALUES
        (seq_test.nextval, ''JOB'' || seq_test.nextval);
    END;',
                                start_date          => to_date('24-08-2019 14:00:00', 'dd-mm-
yyyy hh24:mi:ss'),
                                repeat_interval     => 'Freq=Minutely;Interval=2',
                                end_date            => to_date(null),
                                job_class           => 'DEFAULT_JOB_CLASS',
```

```
                        enabled             => true,
                        auto_drop           => true,
                        comments            => 'FOR TEST');
end;
/

--删除调度


--数据字典视图
SELECT * from user_scheduler_jobs;
SELECT * from User_Scheduler_Job_Log;
SELECT * from user_scheduler_job_run_details;
```

4.7 PACKAGE/PROCEDURE/FUNCTION

高级开发部分

5 复杂查询

```
SELECT * from scott.emp;

SELECT * from scott.dept;

SELECT *
  FROM scott.emp
  JOIN scott.dept
    ON scott.emp.deptno = scott.dept.deptno;

SELECT t1.empno,t1.ename,t1.job,t1.sal,t2.dname "部门名称"
  FROM scott.emp t1
  JOIN scott.dept t2
    ON t1.deptno = t2.deptno ;
```

```
SELECT t1.empno, t1.ename, t1.job, t1.sal, t2.dname "部门名称"
  FROM scott.emp t1, scott.dept t2
 WHERE t1.deptno = t2.deptno;

SELECT * from dept;
SELECT * from emp;

--查找所有部门的员工信息
SELECT *
  FROM scott.dept t1
  LEFT JOIN scott.emp t2
    ON t1.deptno = t2.deptno;

--查找所有员工对应的部门名称
SELECT *
  FROM scott.emp t1
--LEFT JOIN scott.dept t2
  JOIN scott.dept t2
    ON t1.deptno = t2.deptno;

SELECT *
  FROM scott.emp t1
  LEFT JOIN scott.dept t2
    ON t1.deptno = t2.deptno
   AND t2.deptno = 20;
--等价于
SELECT *
  FROM scott.emp t1
  LEFT JOIN (SELECT * FROM scott.dept WHERE deptno = 20) t2
    ON t1.deptno = t2.deptno;

SELECT *
  FROM scott.emp t1
  LEFT JOIN scott.dept t2
    ON t1.deptno = t2.deptno
```

```
    WHERE t2.deptno=20;

CREATE TABLE scott.emp_1 AS
SELECT * from scott.emp;

UPDATE scott.emp_1 SET deptno=50 WHERE sal>2000;
COMMIT;

SELECT deptno from  scott.emp_1
UNION ALL  --不会去掉重复值
SELECT deptno from  scott.emp;

SELECT deptno from  scott.emp_1
UNION   --会去掉重复值
SELECT deptno from  scott.emp;

SELECT t1.empno,
       t1.ename,
       t1.job,
       t1.sal,
       t2.dname "部门名称",
       t3.grade 工资等级
  FROM scott.emp t1
  JOIN scott.dept t2
    ON t1.deptno = t2.deptno
  JOIN scott.salgrade t3
    ON t1.sal BETWEEN t3.losal AND t3.hisal
 WHERE t1.sal > 2000;

 SELECT t1.empno,
        t1.ename,
        t1.job,
        t1.sal,
        (SELECT t2.dname FROM scott.dept t2 WHERE t2.deptno = t1.deptno) 部门名
称,
```

```
       t3.grade 工资等级
  FROM scott.emp t1
  JOIN scott.salgrade t3
    ON t1.sal BETWEEN t3.losal AND t3.hisal
 WHERE t1.sal > 2000;

SELECT * from scott.salgrade;
```

6 单行函数/分析函数

6.1 单行函数(包含聚合函数)

```
SELECT ABS(-1),SQRT(4) from dual ;

-- number(4,2)

SELECT round(192.456), round(192.456, 0), round(192.416), round(192.516)
  FROM dual;

SELECT round(192.456,1), round(192.416,1), round(192.516,2)
  FROM dual;

SELECT TRUNC(192.456), TRUNC(192.456, 0), TRUNC(192.416), TRUNC(192.516)
  FROM dual;

SELECT TRUNC(192.456,1), TRUNC(192.416,1), TRUNC(192.516,2)
  FROM dual;

SELECT ASCII('A')  from dual  ;

SELECT LOWER('Hello WORLD'),  upper('hello world'),LENGTH('hello world') from
dual ;
```

```
SELECT substr('HelloWorld,OCP', 1),
       substr('HelloWorld,OCP', 2),
       substr('HelloWorld,OCP', 1, 2),
       substr('HelloWorld,OCP', -3),
       substr('HelloWorld,OCP', -3, 2)
  FROM dual;

SELECT REPLACE('HelloWorld,OCP', 'o'),
       REPLACE('HelloWorld,OCP', 'o', ''),
       REPLACE('HelloWorld,OCP', 'o', 222)
  FROM dual;

SELECT '   OCP',
       ltrim('   OCP'),
       ltrim('   OCP', ' '),
       ltrim('ooOCPoo', 'o'),
       RTRIM('OCP  '),
       rtrim('ooOCPoo', 'o'),
       TRIM(' OCP ')
  FROM dual;

SELECT nvl(NULL, 'HELLO'),
       nvl('HI', 'HELLO'),
       coalesce(NULL, 'HELLO'),
       coalesce('HI', 'HELLO'),
       NVL2(NULL,'HI','OCP')
  FROM dual;

SELECT SYSDATE,SYSDATE+1,SYSDATE-1,SYSDATE+20/24/60 from dual ;

SELECT SYSDATE, add_months(SYSDATE,2),add_months(SYSDATE,-1) from dual;

SELECT SYSDATE,
       last_day(SYSDATE),
       last_day(add_months(SYSDATE, -1)) + 1,
```

```
       trunc(SYSDATE, 'MM')
  FROM dual;

SELECT SYSDATE,
       trunc(SYSDATE, 'YYYY'),
       trunc(SYSDATE, 'MM'),
       trunc(SYSDATE, 'DD'),
       TRUNC(SYSDATE,'HH'),
       TRUNC(SYSDATE,'MI')
  FROM dual;

SELECT SYSDATE,
       to_char(SYSDATE, 'YYYY-MM-DD hh24:mi:ss'),
       to_date('2019-08-24 16:09:32', 'yyyy-mm-dd hh24:mi:ss')
  FROM dual;

SELECT * FROM scott.emp t WHERE t.hiredate > DATE '1987-04-19';

SELECT Systimestamp, Systimestamp+0 from dual ;

SELECT * from scott.dept;
/*
10    ACCOUNTING
20    RESEARCH
30    SALES
40    OPERATIONS
*/

SELECT decode(t.deptno,
              10,
              'ACCOUNTING',
              20,
              'RESEARCH',
              30,
              'SALES',
```

```
            40,
            'OPERATIONS',
            'NA') 部门名称,t.*
  FROM scott.emp t;

SELECT CASE t.deptno
         WHEN 10 THEN
          'ACCOUNTING'
         WHEN 20 THEN
          'RESEARCH'
         WHEN 30 THEN
          'SALES'
         WHEN 40 THEN
          'OPERATIONS'
         ELSE
          'NA'
       END 部门名称,
       t.*
  FROM scott.emp t;

 SELECT CASE
         WHEN t.deptno=10 THEN
          'ACCOUNTING'
         WHEN t.deptno=20 THEN
          'RESEARCH'
         WHEN t.deptno=30 THEN
          'SALES'
         WHEN t.deptno=40 THEN
          'OPERATIONS'
         ELSE
          'NA'
       END 部门名称,
       t.*
  FROM scott.emp t;
```

```
SELECT Userenv('LANGUAGE') from dual ;

SELECT * from scott.emp;

SELECT COUNT(*),
       COUNT(1),  --查询一张表的总记录数
       COUNT(ename),
       COUNT(comm),
       COUNT(deptno),
       COUNT(DISTINCT deptno)
  FROM scott.emp t;

SELECT MIN(sal),MAX(sal) from scott.emp t
WHERE deptno=20;



--查询工资最高的员工的基本信息
SELECT * FROM scott.emp WHERE sal = (SELECT MAX(sal) FROM scott.emp t);

--获取每个部门的最高工资
SELECT deptno, MAX(sal) sal FROM scott.emp GROUP BY deptno;

--获取每个部门的总工资
SELECT deptno,SUM(sal) from scott.emp
GROUP BY deptno;

--查询每个部门中最高工资的员工基本信息
/*
30    2850
20    3000
10    5000
*/
SELECT *
  FROM scott.emp t1
  JOIN (SELECT deptno, MAX(sal) sal FROM scott.emp GROUP BY deptno) t2
```

```
 ON t1.deptno = t2.deptno
AND t1.sal = t2.sal;
```