1. 数据科学的六层模型
   数据源 - 数据仓库 - 数据探索 - 数据挖掘 - 数据展示 - 做出决策
2. 数据库有哪些核心问题?
   2.1. 数据怎样存储的问题（存储引擎）
   2.2. 数据怎样快速定位的问题（索引）
   2.3. 数据怎样查询和维护的问题（操作命令的解析）
3. 关系型数据库有什么特点？有哪些具体产品?
   关系型数据库，是指采用了关系模型来组织数据的数据库，其以行和列的形式存储数据，以便于用户理解，关系型数据库这一系列的行和列被称为表，一组表组成了数据库。用户通过查询来检索数据库中的数据，而查询是一个用于限定数据库中某些区域的执行代码。关系模型可以简单理解为二维表格模型，而一个关系型数据库就是由二维表及其之间的关系组成的一个数据组织。
   具体产品有**Oracle、DB2、MySQL、Microsoft SQL Server、Microsoft Access**
4. SQL 语言的缘起，有哪些特点?
   4.1. 高级的非过程化编程语言，允许用户在高层数据结构上工作。
   4.2. 不要求用户指定数据存放方法。
   4.3. 不需要用户了解具体数据存放方式。
   4.4. 底层结构完全不同的各种关系型数据库系统可以使用相同的SQL语言作为数据操作和管理的接口。
   4.5. SQL语言可以嵌套，可以通过高级对象实现过程化编程，所以具有很大的灵活性和功能，称为事实上的关系数据库通用语言标准。
5. 熟悉 Oracle 样板表 emp，dept 的结构

```
SQL =  select * from emp;

     EMPNO ENAME      JOB              MGR HIREDATE              SAL       COMM     DEPTNO LOC
---------- ---------- --------- ---------- -------------- ---------- ---------- ---------- ----------
      7369 SMITH      CLERK            7902 17-12月-80           1020                    20 DALLAS
      7499 ALLEN      SALESMAN         7698 20-2月 -81           1980        300         30 CHICAGO
      7521 WARD       SALESMAN         7698 22-2月 -81           1630        500         30 CHICAGO
      7566 JONES      MANAGER          7839 02-4月 -81           3210                    20 DALLAS
      7654 MARTIN     SALESMAN         7698 28-9月 -81           1630       1400         30 CHICAGO
      7698 BLAKE      MANAGER          7839 01-5月 -81           3230                    30 CHICAGO
      7782 CLARK      MANAGER          7839 09-6月 -81           2950                    10 NEW YORK
      7788 SCOTT      ANALYST          7566 19-4月 -87           3210                    20 DALLAS
      7839 KING       PRESIDENT             17-11月-81           5500                    10 NEW YORK
      7844 TURNER     SALESMAN         7698 08-9月 -81           2180          0         30 CHICAGO
      7876 ADAMS      CLERK            7788 23-5月 -87           1310                    20 DALLAS
      7900 JAMES      CLERK            7698 03-12月-81           1630                    30 CHICAGO
      7902 FORD       ANALYST          7566 03-12月-81           3210                    20 DALLAS
      7934 MILLER     CLERK            7782 23-1月 -82           2100                    10 NEW YORK

已选择14行。

SQL =  select * from dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
```

6. Oracle 支持哪些数据类型？怎样观察表列的数据类型

- 字符串类型
- 数字类型

- 日期类型
- LOB类型
- RAW & LONG RAW类型
- ROWID & UROWID类型

```
desc emp;
```

7. select 语句及 where、group by，order by，having 等子句的用途和书写次序

```
select deptno, avg(sal)
from emp
where sal > 2000
group by deptno
having avg(sal) > 3000
order by avg(sal);

    DEPTNO   AVG(SAL)
---------- ----------
        20       3210
        10 3516.66667
```

8. sys 与 system 用户的作用

- sys 最高管理员权限
- system 普通管理员权限
- sys用户具有“SYSDBA”和“SYSOPER”权限，登陆em时也只能用这两个身份，不能用normal。而system登录em时只能用normal模式登录。sys拥有数据字典(dictionary),或者说dictionary属于sys schema。

9. 常用的 sqlplus 变量，它们怎么设置和观察?

```
show all
show linesize
show pagesize
```

10 sql 脚本有什么作用，怎样运行?
批量创建可修改的SQL语句

```
SQL = @DIRECTORY/FILENAME
```

11. 常见 sql 算符

```
select * from emp
where deptno in (10, 20, 30)
and sal between 2000 and 4000
and sal > all(1000, 2300)
and sal < any(3600, 4200)
and ename like 'J%';
```

12. sql 语句中的逻辑组合计算

```
select * from emp
where (
    comm is null and sal > 3000
) or (
    comm is not null and sal + comm > 3000
);
```

13. 怎样创建与删除表

```
create table account(
    ID, number(10), primary key,
    PWD, varchar2(24), not null,
    EMAIL, varchar2(100), unique,
    QQ, number(13),
    constraint QQCST foreign key (QQ) references QQLST(ID)
);

drop table account;
```

14. Oracle 支持哪些约束？每种约束的含义。创建、删除、关闭或打开约束的操作

- 主键约束（primary key）
- 外键约束（foreign key）
- 唯一性约束（unique）
- 非空约束（not null）
- 检查约束（check）

```
alter table sc disable constraint QQCST;
```

15. 怎样增删表列?

```
alter table user_info add email varchar2(100) unique;
```

16. SQL 函数

```
-- 聚组函数
avg(), sum(), count(), max(), min(), stddev(), variance(),
-- 用 t 补齐 s 的左边，使长度为 n
lpad(s:str, n:num, t:str)
-- 用 t 补齐 s 的右边，使长度为 n
rpad(s:str, n:num, t:str)
-- 第一个字符大写，其他字符小写
initcap(lower(s:str))
-- 取子串
substr(s:str, init:num, len:num)
-- 从第一个不在 t 中的字符开始一直取到 s 结尾
ltrim(s:str, t:str = ' ') -> str
ltrim('abcdefg', 'badf') = 'cdefg'
-- 从右边开始取
rtrim(s:str, t:str = ' ') -> str
-- 返回t中任意字符在s中出现的第一个位置
instr(s:str, t:str) -> num
-- ASCII转字符
chr(n:num) -> str
-- 字符串替换
replace(s:str, origin:str, alt:str) -> str
-- 按表替换
translate(s:str, origin:str, alt:str) -> str
translate('abcde', 'ace', '123') = '1b2d3'
-- 按读音查询
soundex(s:str) -> str
soundex('scott') = soundex('skot')
-- 替换 null
nvl(col:str, n:any) -> any
-- 向上取整
ceil(n:num)
-- 向下取整
floor(n)
-- 四舍五入，并保留t位小数
round(n:num, t:num)
-- 截断，保留t位小数
trunc(n:num, t:num)
-- 取最大值
greatest(1, 2, 3, 4) = 4
-- 取最小值
least(1, 2, 3, 4) = 1
-- 转字符串
to_char(n:num, format:str)
to_char(123.45, '$0099.99') = '$0123.45'
to_char(d:date, format:str)
to_char(sysdate, 'yyyy-mm-dd hh:mi:ss')
-- 转数字
to_number(s:str)
-- 转日期
to_date(s:str, format:str)
-- 获取当前用户
select user from dual;
```

17. sqlldr 的用法，控制文件的写法

```
OPTIONS (skip=0)
load data
INFILE  'score_data.csv'
insert into table score
FIELDS TERMINATED BY ', '
trailing nullcols (
    SID,
    CID,
    SCORE
)
```

```
sqlldr scott/tiger control=Load.ctl
```

18. 有哪些伪列函数，分别有什么作用?

```
rownum, rowid
```

19 什么是视图？视图有什么作用和操作特点，创建与删除视图
视图即查询语句的组合。作用在于：

- 简化sql书写
- 隐蔽数据细节
- 实现行列级别的安全控制
- 解决top n问题

```
create view v10t as
select ename, sal + nvl(comm, 0) tal
from emp
where deptno = 10;

drop view v10t;
```

20. insert，delete，update 等 DML 操作

```
insert into user_info values(
    'Hello', 12345, 'C'
);
delete user_info where email = 'C';

update user_info
set passwd = passwd + 1;
```

21 什么是事务及其 ACID 特性?

- 原子性（Atomicity）：即不可分割性，事务中的操作要么全不做，要么全做
- 一致性（Consistency）：一个事务在执行前后，数据库都必须处于正确的状态，满足完整性约束
- 隔离性（Isolation）：多个事务并发执行时，一个事务的执行不应影响其他事务的执行
- 持久性（Durability）：事务处理完成后，对数据的修改就是永久的，即便系统故障也不会丢失

22. 什么是事务的隔离性？具体有哪些，分别实现什么效果?

- 读未提交（READ UNCOMMITTED）
- 读提交 （READ COMMITTED）
- 可重复读 （REPEATABLE READ）

- 串行化 （SERIALIZABLE）

隔离级别	脏读	不可重复读	幻读
读未提交	可能	可能	可能
读提交	不可能	可能	可能
可重复读	不可能	不可能	可能
串行化	不可能	不可能	不可能

> 脏读
> 脏读指的是读到了其他事务未提交的数据，未提交意味着这些数据可能会回滚，也就是可能最终不会存到数据库中，也就是不存在的数据。读到了并一定最终存在的数据，这就是脏读。

> 可重复读
> 可重复读指的是在一个事务内，最开始读到的数据和事务结束前的任意时刻读到的同一批数据都是一致的。通常针对数据更新（UPDATE）操作。

> 不可重复读
> 对比可重复读，不可重复读指的是在同一事务内，不同的时刻读到的同一批数据可能是不一样的，可能会受到其他事务的影响，比如其他事务改了这批数据并提交了。通常针对数据更新（UPDATE）操作。

> 幻读
> 幻读是针对数据插入（INSERT）操作来说的。假设事务A对某些行的内容作了更改，但是还未提交，此时事务B插入了与事务A更改前的记录相同的记录行，并且在事务A提交之前先提交了，而这时，在事务A中查询，会发现好像刚刚的更改对于某些数据未起作用，但其实是事务B刚插入进来的，让用户感觉很魔幻，感觉出现了幻觉，这就叫幻读。

23. 什么是提交与回滚，各有什么作用。显式和隐式操作
    事务提交就结束事务，回滚将回到事务开始之前

```
commit;
rollback;
```

24. 回滚段怎样实现读一致性，回滚和提交操作与回滚段有什么关系?

查询时碰到数据修改时间戳晚于查询开始时间，则在回滚段进行查找。

25. 笛卡尔积，连接，内连接，左外连接，右外连接，全外连接等多表操作

```
select a, b, c from (
    select e1.ename a, e2.ename b, e1.deptno c
    from emp e1, emp e2
) full outer join dept
on c = dept.deptno
where b = 'SMITH';
```

26. 子查询

```
select * from (select * from emp);

select sn from s
where not exists (
    select * from c
    where not exists (
        select * from sc
        where s# = s.s# and c# = c.c#
    )
);
```

27. 聚组统计的实现，聚组函数，rollup，cube，grouping

- rollup: 按顺序数据小计

```
select dname, ename, sum(sal + nvl(comm, 0)),
grouping(ename)
from emp
left join dept on emp.deptno = dept.deptno
group by rollup (dname, ename);

DNAME          ENAME      SUM(SAL+NVL(COMM,0)) GROUPING(ENAME)
-------------- ---------- -------------------- ---------------
SALES          WARD                       2130               0
SALES          ALLEN                      2280               0
SALES          BLAKE                      3230               0
SALES          JAMES                      1630               0
SALES          MARTIN                     3030               0
SALES          TURNER                     2180               0
SALES                                    14480               1
RESEARCH       FORD                       3210               0
RESEARCH       ADAMS                      1310               0
RESEARCH       JONES                      3210               0
RESEARCH       SCOTT                      3210               0
RESEARCH       SMITH                      1020               0
RESEARCH                                 11960               1
ACCOUNTING     KING                       5500               0
ACCOUNTING     CLARK                      2950               0
ACCOUNTING     MILLER                     2100               0
ACCOUNTING                               10550               1
                                         36990               1
```

- cube: 对rollup排列组合
- grouping: 判断是否为null

28. decode 函数与 case 语句

```
sal = sal + decode(
    loc, 'NEW YORK', 300,
    loc, 'DALLAS', 150
);
case
when ... then ...
when ... then ...
end
```

29. with 语句的用法

```
-- 没什么用
with tmp as (select * from emp)
select * from tmp;
```

30. 注释语句
31. 什么是宽表与窄表？怎样使用 pivot 和 unpivot 函数实现宽窄表互相转化？

```
SELECT * FROM (
    select * from score
    pivot(
        sum(score)
        for cid in (
            'C1' C1, 'C2' C2, 'C3' C3, 'C4' C4, 'C5' C5
        )
    )
) unpivot (score for cid in (
    "C1", "C2", "C3", "C4", "C5"
));
```

32. 集合运算有哪些?

```
intersect, minus, union
```

33. 表（关系）除法的意义，怎么实现？exists 与 exists 算符
34. merge 语句的用法

```
merge into origin
use new
on (origin.item = new.item)
when matched then
    update set origin.item1 = new.item1
when not matched then
    insert values(
        new.item, new.item1
    );
```

35. 层次查询及其执行原理

```
select ename from emp
start with ename = 'SMITH'
connect by prior mgr = empno;
```