**考点8：数据库相关概念（1%）**

**数据库、数据库管理系统与 SQL 之间的关系（掌握）**

数据库是存储、调用、分析数据的仓库，是带有相关数据的表的集合，分类：关系型数据库，非关系型数据库

数据库基本结构：1、表：一个表是多个字段的集合 2、字段：一个字段是一列数据，由字段名和记录组成

关系型数据库管理系统RDBMS：数据库应用主流，以行和列的形式存储数据表，一组表组成了数据库，如Oracle、DB2、MySQL

SQL结构化查询语言：操作关系型数据库时使用的语言

**考点9：DDL 数据定义语言（2%）**

数据定义语言DDL：用来对数据库管理系统中的对象进行“增删改查”操作的SQL语句，一般应用DDL语言对数据库不同对象进行定义操作时，需要提前具备相应数据库管理权限

**针对数据库对象的操作语言**

创建数据库：create database test；查看创建好的数据库：show create database test；查看所有数据库列表：show databases test;

使用数据库：use test； 删除数据库：drop database test；

**针对数据表对象的操作语言**

数据库由多个数据表构成，每张数据表存储多个字段，每个字段由不同的字段名及记录构成，每个字段都有相应的数据结构及约束条件

创建数据表：create table emp (id char(3), name varchar(20));

查看表是否创建成功：show tables;

删除数据表：drop table emp;

修改表名：alter table emp rename empdep;

修改字段类型：alter table empdep modify depname varchar(30);

修改字段名：alter table empdep change depname dep varchar(20);

增加字段：alter table empdep add newname varchar(10) not null;

删除字段：alter table empdep drop newname;

修改字段的排列位置：modify ...first排序改为第一位；modify...after...排列顺序改到某个字段之后

alter table empdep modify newname varchar(10) first;

alter table empdep modify newname varchar(10) after depid;

**数据类型（掌握）**

不同的数据类型具有不同的字节长度，字节长度越长能取值的数值区间也就越大，1字节=8二进制位

数据表中的数据类型：

1、整数：INT、TINYINT、SMARTINT、MEDUMINT、BIGINT，若加上UNSIGNED可以禁止负数

2、小数（只能为有符号的）：Float四字节，double八字节，decimal

3、日期和时间：date，datetime，time，timpstamp，year

4、字符串：char()固定长度字符串；varchar()可变长度字符串；BOLB或TEXT

**约束条件（掌握）**

是在表上强制执行的数据检验规则，用来保证创建的表数据完整和正确，MySQL数据库常用约束条件：

1、主键约束primary key：保证表中每行记录都不重复。主键列的数据必须唯一，且不允许为空

主键类型：单字段主键depid char(3) primary key，多字段联合主键primary key(depname,depid)

2、非空约束not null：字段的值不能为空，语法：depname varchar(20) not null

3、唯一约束unique：该列的值必须唯一，语法：peoplecount int unique

①允许为空，但只能出现一个空值

②一个表中可以有多个字段声明为唯一的

③唯一约束确保数据表的一列或几列不出现重复值；

4、默认约束default ：指定某个字段的默认值，语法：depname varchar(20) default ‘-’；

5、自增字段：一个表只能有一个自增字段，且必须为主键的一部分，默认情况下从1开始自增，语法id int primary key auto\_increment

**考点10：DML 数据操作语言（2%）**

数据操作语言DML：对表中记录进行添加（INSERT）、更新（UPDATE）、删除（DELETE）等操作的语言，需要拥有相应的操作权限

**添加INSERT（掌握）**

INSERT INTO fruit(f\_id, s\_id, name, f\_price) VALUES ('a1', '101', 'apple', 5.2), ('b1', '102', 'banana', 10.3)

**导入外部文本文件：load**

检查导入内容：select \* from 表名；

检查导入数据总行数：select count（\*） from 表名；

检查表结构：desc 表名；

**为字段赋值UPDATE 表名 SET 字段名=值（掌握）**

例如：使用concat函数在f\_name字段前添加‘fruit\_’信息

UPDATE fruits SET f\_name = concat(‘fruit\_’, f\_name)

**删除记录DELETE（掌握）**

例如：删除f\_id为'b5'的数据记录

delete from fruits where f\_id = 'b5'

**考点11：单表查询（3%）**

**运算符（掌握）**：加+ 减- 乘\* 除/ 等于= 大于> 小于< 大于等于>= 小于等于<= 不等于!= <> 不大于!> 不小于!<

聚合函数：按列算平均值AVG，按列求和SUM，求1列最大值MAX，求1列最小值MIN，按列计个数COUNT

**SELECT查询（应用）**

SELECT 目标列组 FROM 数据源 WHERE 选择条件

GROUP BY 分列组 HAVING 组选择条件 ORDER BY 排序列 排序要求；

-- 对大气质量表进行有选择的查询

select city\_name AS 城市, avg(pm25), avg(pm10) from Monthly\_Indicator

where pm25 > 50

group by city\_name, month\_key having city\_name <> '北京'

order by avg(pm25) desc;

**考点12：多表查询（3%）**

多表查询：将两个以上的数据表通过关键字段连接在一起，并从不同表中取不同字段进行查询

关键字段：用来连接两表的内容信息能够匹配的上的字段

1、相连的两表中都需要有关键字段

2、关键字段中的记录信息能够匹配得上

3、最理想的连接状态是两表中的两个关键字段都是主键，而且两个主键的值能够一一匹配的上

4、语法：select <select\_list> from 表1 <left/right/inner> join 表2 on 表1.key=表2.key

**内连接**inner join**（掌握）**：按照连接条件合并两个表，返回满足条件的行

Select <select\_list> from A INNER JOIN B on A.key=B.key;

**左连接**left join**（掌握）**：结果中除了包括满足连接条件的行外，还包括左表中的所有行

Select <select\_list> from A left join B on A.key=B.key;

**右连接**right join**（掌握）**：结果中除了包括满足连接条件的行外，还包括右表中的所有行；

Select <select\_list> from A right join B on A.key=B.key;

**联合查询（掌握）**

UNION：用于合并两个或多个select语句的结果集，去重

UNION ALL：用于合并两个或多个select语句的结果集，不去重

select t1.\* from t1 union/union all select t2.\* from t2;

全连接查询：全连接会返回两张表中全部的记录，本质上是两个表的并集

全连接没有主附表的区别，按照连接条件可以匹配到的记录会返回匹配后的结果，匹配不到的记录用null进行填充

Select \* from t1 left join t2 on key1=key2 Union Select \* from t1 right join t2 on key1=key2;

**考点13：子查询（3%）**

**操作符（掌握）**

AND：且。条件表达式1 and 条件表达式2...and 条件表达式n

OR：或。条件表达式1 or 条件表达式2...or 条件表达式n

IN：判断某个字段的值是否在指定的集合中，如果在集合中则满足查询条件，如果不在则不满足查询条件。in/not in（元素1，元素2，...元素n）

BETWEEN：判断某个字段的值是否在指定的范围内，如果在则满足查询条件，如果不在则不满足查询条件。【not】 between 取值1 and 取值2

LIKE：用来匹配字符串是否相等，如果字段的值与指定的字符串相匹配，则满足查询条件，如果与指定的字符串不匹配，则不满足查询条件

[not] like 字符串，字符串参数的值可以是一个完整的字符串，也可以是包含%或者\_的通配符，其中%代表任意长度的字符串，\_代表单个字符

IS NULL：用来判断字段的值是否为空值null，如果字段的值为空值，则满足查询条件，如果字段的值是非空值，则不满足查询条件Is [not] null

DISTINCT：用来消除重复记录。Select distinct 字段名

**操作符与子查询的组合应用（掌握）**

子查询：写在（）中，把内层查询结果当做外层查询参照的数据表来用

ANY：满足内层查询语句返回的结果中的任何一个，就可以通过该条件来执行外层查询语句

ALL：满足内层查询语句返回的所有结果，才可以执行外层查询语句

EXIST：内层查询语句不返回查询的记录，而是返回一个真假值，

如果内层查询语句查询到满足条件的记录，就返回真值，外层查询语句将进行查询，否则返回假值，外层查询语句不进行查询；

As重命名：可以将表或字段名重新命名为别的名称使用，只在查询中有效；

Limit：查询后只显示limit指定数字的行数结果；

**考点14：数据库函数（3%）（应用）**

**数学函数**

ABS(x)：返回x的绝对值

LOG(x,y)：返回x的以y为底的对数

MOD(x,y)：返回x/y的模（余数）

RAND()：返回0到1内的随机值，可以通过提供一个参数（种子）使RAND()随机数生成器生成一个指定的值

CEILING(x)：返回大于x的最小整数值（进一取整）

ROUND(x,y)：返回参数x的四舍五入的有y位小数的值（四舍五入）

SQRT(x)：返回一个数的平方根

-- 使用abs函数求所有水果平均值与最大值差值的绝对值

select abs(avg(f\_price)-max(f\_price)) from fruits;

**字符串函数**

CONCAT(s1,s2,...,sn)：将s1,s2,...,sn拼接成字符串

LEFT(str,x)：返回字符串str中最左边的x个字符

RIGHT(str,x)：返回字符串str中最右边的x个字符

LENGTH(str,x)：返回字符串str中的字符数

-- 使用length函数求每个f\_name的名字与他们的字符长度

select f\_name, length(f\_name) from fruits group by f\_name;

**日期及时间函数**

NOW()：返回当前的日期和时间

DATE(datetime)：返回datetime的日期值

TIME(datetime)：返回datetime的时间值

-- 使用now函数求当前的日期和时间

select now();

**分组合并函数**

GROUP\_CONCAT(col)：常与关键字GROUP BY一起使用，返回由属于一组的列值连接组合而成的结果

CAST()：将一个值转换为指定的数据类型

-- 示例：使用GROUP\_CONCAT()函数查询不同s\_id对应的所有f\_name信息

SELECT s\_id, GROUP\_CONCAT(f\_name)

FROM fruits

GROUP BY s\_id;

-- 示例：使用CAST()函数，将文本字符串'12'转换为整型12

SELECT CAST('12' AS INT);

**逻辑函数**

### **空值函数**：IFNULL(expr, value)：如果expr值为NULL，则替换为value

### **判断函数（单条件）**：IF(expr1,expr2,expr3)：expr1判断，真2，假3

### **逻辑表达式（多条件）**：CASE WHEN expr1 THEN expr2 [when expr3 then expr4 ... else expr] end;

-- 空值函数IFNULL(expression, alt\_value)  
-- 查询每位员工实发工资（基本工资+奖金）  
select ename, sal + ifnull(comm, 0) 实发工资 from emp;  
​  
-- IF判断  
select ename, sal, IF(sal>=3000, '高', IF(sa<=1500, '低', '中')) 工资级别 from emp;  
​  
-- CASE WHEN ... THEN ... ELSE ... [WHEN ... THEN ... ELSE] ... END  
select ename, sal, case when sal>=3000 then '高' when sal<=1500 then '低' else '中' end 工资级别 from emp;