

# ACA P41 - P65 SQL基础

- **SQL基础 (P41)**

- **1. SQL简介 (P42-P44)**

- 结构化查询语言(Structured Query Language), 简称SQL, 是数据库编程的核心语言, 是数据分析的必备工具
      - 一体化 SQL 集数据定义 DDL、数据操纵DML和数据控制 DCL 于一体, 可以完成数据库中的全部工作
      - 使用灵活 有两种使用方式, 即可以直接以命令方式交互使用; 也可以嵌入使用, 嵌入到C、JAVA等主语言中使用
      - 非过程化只提操作要求, 不必描述操作步骤, 也不需要导航。使用时只需要告诉计算机做什么, 而不需要告诉它怎么做
      - 简单易用简洁、简单, 易学好用, 在 ANSI标准中, 只包含了 94 个英文单词, 核心功能只用 6 个动词, 语法接近英语口语

- **SQL分类 SQL语言有以下三大类**

- 数据查询语言DQL
    - 数据操纵语言DML
    - 数据定义语言DDL

- **SQL与数据分析 • 常见应用方式:**

- 常见应用方式:
      - 关系型数据库管理系统RDBMS, 如MySQL
      - Hadoop大数据平台, 如Hive
      - 阿里云大数据平台MaxCompute, 相对于Hadoop平台, 成本低、更易用、功能更强大、更安全
    - 数据分析
      - 小规模一般统计分析, 通过客户端工具将数据导入到MySQL, 使用SQL语言及内置函数实现数据清洗、加工、统计
      - 海量数据离线分析,通过Sqoop等工具将数据库数据导入Hive, 通过Hive SQL完成数据清洗、加工、统计
      - 海量数据离线分析,通过数据集成、客户端等工具将数据同步到MaxCompute。通过DataWorks开发, MaxCompute SQL完成数据清洗、加工、统计

- **SQL与RDBMS**

- 关系数据库管理系统 (RDBMS) 全称Relational Database Management System, 是为管理数据库而设计的软件系统, 一般具有存储、截取、安全保障、备份等基础功能

- 数据模型是数据库系统的核心和基础；借助于集合代数等数学概念和方法来处理数据，包括选择、投影、连接、并、交、差、除、增删查改等实现对数据的存储和查询；
  - 一个关系型数据库就是由二维表及其之间的联系组成的一个数据组织，通过SQL实现数据操作；
  - 关系数据库使用主外键机制维护维护关系数据库的完整性；
  - 相对于以往的文件系统，数据库系统使用更方便；相对于其他模型数据库关系模型更贴近逻辑世界。
- 2. SQL基础操作（P45）
    - 2.1 DDL操作
      - DDL综述
        - DDL(Data Definition Language): 数据定义语言，用于创建、修改、删除数据库中的对象：
        - CREATE: 创建
        - ALTER: 修改
        - DROP: 删除
        - TRUNCATE 清空操作
        - COMMENT 备注处理
        - RENAME：更名操作
      - DDL建库（P46）
        - 使用create database或create schema命令可以创建数据库。其语法结构如下：
          - create {database|schema}[if not exists]databasename
          - [default]character set charset\_name
          - [[default]collate collation\_name;
          - 【例】创建一个名称为mysqltest的数据库。
          - create database if not exists mysqltest;
        - 在创建数据库时，数据库命名有以下几项规则（MySQL为例）：
          - 不能与其他数据库重名，否则将发生错误。
          - 名称可以由任意字母、阿拉伯数字、下划线（\_）和“\$”组成，可以使用上述的任意字符开头，但不能使用单独的数字，否则会造成它与数值相混淆。
          - 名称最长可为64个字符，而别名最多可长达256个字符
          - 不能使用关键字作为数据库名、表名。
          - Version:0.9StartHTML:0000000105EndHTML:0000000660StartFragment:0000000141EndFragment:0000000620
          - 说明：在默认情况下，Windows下数据库名、表名的大小写是不敏感的，而在Linux下数据库名、表名的大小写是敏感的。如果为了便于

数据库在平台间进行移植，可以采用小写来定义数据库名和表名。

- **DDL建表 (P47)**

- 创建表的语法结构：
  - create [temporary]table[if not exists]table\_name
  - [[column\_definition], ...][index\_definition]]
  - [table\_option][select\_statement];
- column\_definition: 字段的定义，包括指定字段名、数据类型、是否允许空值，指定默认值、主键约束、唯一性约束、注释字段名、是否为外键，以及字段类型的属性等。字段定义具体格式如下：
  - col\_name type [not null | null] [default default\_value]
  - [auto\_increment] [unique [key] | [primary] key]
  - [comment 'string'] [reference\_definition]
- 创建表示例：
  - create table `t\_test\_tmp` (
  - `id` bigint(20) not null auto\_increment,
  - `app\_id` varchar(128) not null comment '公众号app\_id',
  - primary key (`id`),
  - unique key `t\_test\_tmp` (`app\_id`)
  - ) comment= '公众号数据登记表';

- **DDL修改数据表**

- 修改表的语法结构：
  - alter [ignore] table tbl\_name
  - alter\_specification [, alter\_specification] ...
  - alter\_specification:
    - add [column] column\_definition [first | after col\_name] //添加字段
    - |alter [column]col\_name{set default literal|drop default} //修改字段默认值
    - |change [column] old\_col\_name column\_definition //重命名字段
    - [first|after col\_name]
    - |modify [column]column\_definition[first|aftercol\_name] //修改字段数据类型
    - |drop [column] col\_name //删除列
    - |rename [TO] new\_tbl\_name //对表重命名
    - |order by col\_name //按字段排序
    - |convert TO character set charset\_name[collate collation\_name] //将字符集转换为二进制

- [[default] character set charset\_name [collate collation\_name]]//修改表的默认字符集
- 修改表用于修改原有表的结构信息：
  - 增加字段。在创建表时，表中的字段就已经定义完成；如果要增加新的字段，可以通过alter table语句进行增加
  - 修改表名。表名可以在一个数据库中唯一的确定一张表。数据库系统通过表名来区分不同的表
  - 修改字段的数据类型。alter table语句也可以修改字段的数据类型
  - 删除字段。删除字段是指删除已经定义好的表中的某个字段
  - 修改备注信息
- 【例】 在student表的Email列后面增加一列address。
  - alter table student add address varchar(30) not null after Email;
- **DDL其他操作•删除数据库、删除表、清空表等**
  - 删除数据库语法格式如下：
    - drop database [if exists] db\_name
  - 删除表、清空表的语法格式如下：
    - drop table table\_name ;
    - truncate table table\_name ;
  - 操作示例
    - drop database mysqltest; ----删除数据库
    - drop table `t\_test\_tmp` ; ----删除表
    - truncate table `t\_test\_tmp` ; -----清空表
- 2.2 DML操作（P49）
  - DML(Data Manipulation Language): 数据操纵语言，用于对表中的数据进行新增、删除、修改操作，
    - INSERT: 插入
    - DELETE: 删除
    - UPDATE: 更新
  - **DML插入数据**
    - insert into 语句添加数据，语法格式：
      - insert [into]table\_name[(col\_name,...)]
      - values({expr|default},...),(...),...
      - |set col\_name={expr|default}, ...
    - 说明
      - 使用insert语句可以向表中插入一行数据，也可以插入多行数据
      - 一次插入多行数据，各行数据之间用“，”分隔

- values子句包含各列需要插入的数据清单，数据的顺序要与列的顺序相对应
- 若表名后不给出列名，则在values子句中要给出每一列的值，如果列值为空，则值必须置为null，否则会出错
- 操作示例：
  - Insert into  
`dim\_product`(`product\_id`,`product\_name`,`price`,`product\_category\_id`) values('1','信封',25.83,'1'),('2','剪刀',42.25,'1');

## • DML修改数据

- update...set... 命令可以修改一个表的数据，语法格式：
  - update table\_name
  - set col\_name1=[,col\_name2=expr2 ...] [where子句]
- 说明：
  - set子句：根据where子句中指定的条件，对符合条件的数据行进行修改。若语句中不设定where子句，则更新所有行
  - expr1、expr2.....可以是常量、变量或表达式。可以同时修改所在数据行的多个列值，中间用逗号隔开。
- 【例】将学号为18137221508学生的课程号为 c08106的平时成绩daily修改为80分。
  - update score set daily=80 where studentno='18137221508' and courseno='c08106';

## • DML删除数据

- 利用delete... from...语句可以从单个表中删除指定表数据，语法格式：
  - delete [low\_priority] [quick] [ignore] from tbl\_name[where子句] [order by子句] [limit row\_count]
- 说明：
  - quick修饰符：可以加快部分种类的删除操作的速度。
  - from子句：用于指定从何处删除数据。
  - where子句指定的删除条件。如果省略where子句则删除该表的所有行。
  - order by子句：各行按照子句中指定的顺序进行删除，此子句只在与limit联用时才起作用。
  - limit子句：用于告知服务器在控制命令被返回到客户端前被删除的行的最大值。
- 【例】删除student01表中入学成绩低于750分记录。
  - delete from student01 where entrance <750;

## • 2.3 DQL操作

### • DQL综述

- DQL(Data Query Language): 数据查询语言, 用于查询中的数据, SELECT 语句提供了字段运算、数据筛选、分组聚合、多表关联等功能, 主要涉及: SELECT、FROM、WHERE、ORDER BY、GROUP BY、HAVING 等子句。

- 相关操作如下:

- □ 查询 (SELECT)
- □ 别名 (AS)
- □ 去重 (DISTINCT)
- □ 过滤 (WHERE)
- □ 集合判断 (IN)
- □ 范围条件 (BETWEEN AND)
- □ 模糊匹配 (LIKE)
- □ 空值查询 (IS NULL)
- □ 多条件查询 (AND/OR)
- □ 排序 (ORDER BY)
- □ 分组 (GROUP BY & HAVING)
- □ 数量限制 (LIMIT)
- □ 分支判断 (CASE WHEN)

- **DQL查询select**

- • select 语句是SQL语言从数据库中获取信息的一个基本语句:

- select [all|distinct]selection\_list
- from table\_source
- [where search\_condition]
- [group by grouping\_columns][with rollup]
- [having search\_condition]
- [order by order\_expression [asc|desc]]
- [limit count]

- • 说明:

- • 可以实现从一个或多个数据库中的一个或多个表中查询信息
- • 查询结果 (也称结果集) 显示为另外一个二维表的形式

- 注意实例

- (1) 使用select语句查询一个数据表。使用select语句时, 首先要确定所要查询的列。“\*”代表所有的列。
  - 【例】查询teaching数据库course表中的所有数据。
  - select \* from course;
- (2) 查询表中的指定列。针对表中的多列进行查询, 只要在select后面指定要查询的列名即可, 多列之间用“, ”分隔。
  - 【例】查询student表中的studentno、sname和phone数据。

- select select studentno,sname,phone from student;  
studentno,sname,phone from student;
- (3) 可以从一个或多个表中获取数据，即多表查询，主要采用多表连接或子查询的方式，也可以通过where子句中使用连接运算来确定表之间的联系，然后根据这个条件返回查询结果。
  - 【例】 student表、course表关联查询学生上课的信息。
  - select select a.studentno,a.sname,b.cname from student a , course b where a. studentno=b. studentno;

- Version:0.9StartHTML:0000000105EndHTML:0000000337StartFragment:0000000141EndFragment:0000000297

- **DQL别名as、去重distinct**

- 

- 3. SQL高级应用

以上内容整理于 [幕布文档](#)