

数据库使用最多的是 mysql 数据库，在整个大学期间到现在都有使用。

(DDL) :

掌握数据库基本的表创建，主外键创建，字段索引创建，值约束等。根据系统业务需求设计合理的数据表。根据字段选择合适的数据类型，主键 id 使用 int 或者 bigint，带小数的数据使用 float 或 double，字符串使用 varchar，text 等，爱好之类的可以用 set，性别可以用 emun，每日签到使用布尔 (tinyint)。

主键用于每条数据的唯一标识。比如在用户信息表中，可以将 user_id 字段设置为主键并设置为自增。

外键的可以将两个表建立联系同时对该字段进行约束，比如有个用户表和订单表，这两个表相关联的，在订单表中的 user_id 字段设置为外键，它的值依赖于用户表中的 user_id 的值。

索引的使用，如果对用户表的用户名字段频繁的查询，表中的数据也非常的庞大，就可以为用户名字段创建一个普通的索引 (单列索引)，提高查询效率，虽然该表的 DML 操作性能可能下降，但是在这里影响不大，因为用户的注册，删除，修改可能没那么的频繁。

(DML) :

掌握基本的 sql 语句的增删改，使用 insert into values 在指定表中插多条数据。delete from where 从表中删除符合条件的语句。update set where 语句用于更新某表指定字段的值。delete 和 updata 都必须写合理的 where 条件，否则将删除或更新整张表。

(DQL) :

简单的查询语句比如：select *或一些字段名 from 表名 where 加一定的条件，比如 age>15 and name LIKE “李%” 进行简单查询。常用的聚合函数使用 count(), sum(), avg(), max(), min()等。

会使用基本的嵌套查询，例如在 select, from 或 where 中嵌套子查询。常用的操作符：>, <, =, in, not in, 表示是否在子查询结果中，exists, not exists 表示子查询是否有查询结果。

使用 group by 根据字段分组，例如 select city,count(*) from user group by city,就可以查询对应城市有多少人。

利用 Having 对分的组进行过滤，例如 having count(*)>10, 过滤出对应城市人数大于 10 的城市。having 语句必须跟在 group by 语句之后使用，因为 having 是对分组后的结果进行筛选。如果没有 group by 语句，单独使用 having 是没有意义的。但是在隐式分组的情况下是可以使用 Having 的，例如：select avg(amount) from order having avg(amount) >1000

利用 order by 字段名 DESC 或 ASC 进行降序或升序排序。Limit 的使用，limit 5 就是取查询结果的前 5 条，limit 5,8. 就是忽略查询结果的前 5 条，取 8 条数据。利用 limit 就可以实现分页查询的功能。

还有表连接查询，内连接：select 字段名 from tabA inner join tabB on tabA.id = tabB.id, 返回两个表满足条件的行。内连接可以用在查询用户有哪些订单的情况两表之间有一定的联系。左连接和右连接的使用都是分别用 left 或 right 替换掉 inner 语句。左连接

就是返回左表所有行，即使右表没有匹配的行。右连接就是返回右表所有行，即使左表没有匹配的行。左连接的特点是保证了左表数据的完整，右连接很少使用，甚至可以用左连接替代。内连接，左连接，右连接可以根据实际业务需求选择。

还有视图的创建，使用 `create view` 视图名 `as` 后面写一些 DQL 语句，视图使用起来就像是去查询了一个不存在的表，但是得到的数据都是有效的。使操作简单化，对一些字段重新取名字，保证数据的安全性。

存储过程

掌握存储过程的使用，存储过程类似于一个函数，将流程复杂的 sql 语句操作封装起来，在创建后可以反复的调用，存储过程也能输入参数和输出参数，它是使用 IN 或者 OUT 加上参数名加上参数类型完成，配合 `declare`，`while`，`if` 等流程控制语句，实现更复杂，能复用的，使用简单的存储过程。调用存储过程使用 `call` 加存储过程名。

触发器

还有掌握触发器的使用，触发器是特殊的存储过程，触发器的触发条件可以是 `insert`，`update` 和 `delete`。触发方式有 `after` 和 `before`，分别是在事件之前或之后触发。触发器是被动的，满足触发条件时自动触发，不能像存储过程那样使用 `call` 主动触发。

事务

还有事务的使用，开启一个事务可以使用 `START TRANSACTION` 或 `begin` 语句。在事务中，可以执行多个 SQL 语句，如插入、更新或删除操作。事务的结束由 `Commit` 或 `RollBack` 命令来控制。`Commit` 语句用于保存本次事务中 sql 语句执行的结果，使用 `Rollback` 来回滚操作。事务还有一个特点就是：如果事务中有任何一条 sql 语句执行失败，将会自动回滚到开启事务之后的状态。而且多个事务可以并发执行，互不干扰，这就是事务最大的用处。