# 数据库技术知识点手册

## Chapter1 数据库概念

### 一、数据与数据库（理解）

**数据**：是客观事物的反映和记录，是用以载荷信息的物理符号。数据是对客观事物记录下来的可以鉴别的符号。这些符号不仅指数字，而且包括字符、文字、图形等。可以把数据理解为书中用于描述事物的符号。

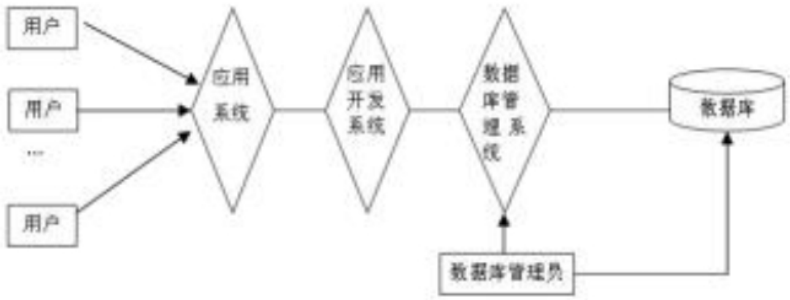
**数据库**：所谓数据库(Database，DB)，是将数据按一定的数据模型组织、描述和存储，具有较小的冗余度，较高的数据独立性和易扩展性，并可为各种用户共享的数据集合。

### 二、数据库技术的发展

**人工管理阶段-〉文件系统阶段 -〉 数据库管理（特点）**

### 三、数据库系统的组成

**数据库系统**(Database System，DBS)一般由**数据库**（DB）、**数据库管理系统**(及其开发工具，DBMS)、**应用系统**、**数据库管理员**和**用户**组成。




### 四、结构化查询语言 SQL

**SQL，structed Qurey Language** 即**结构化查询语言**，数据库管理系统通过 sql 语言来管理数据库中的数据。SQL 语言集数据查询(data

query)、数据操纵(data manipulation)、数据定义(data definition)和数据控制（data control）功能于一体，充分体现了关系数据语言的特点和优点。

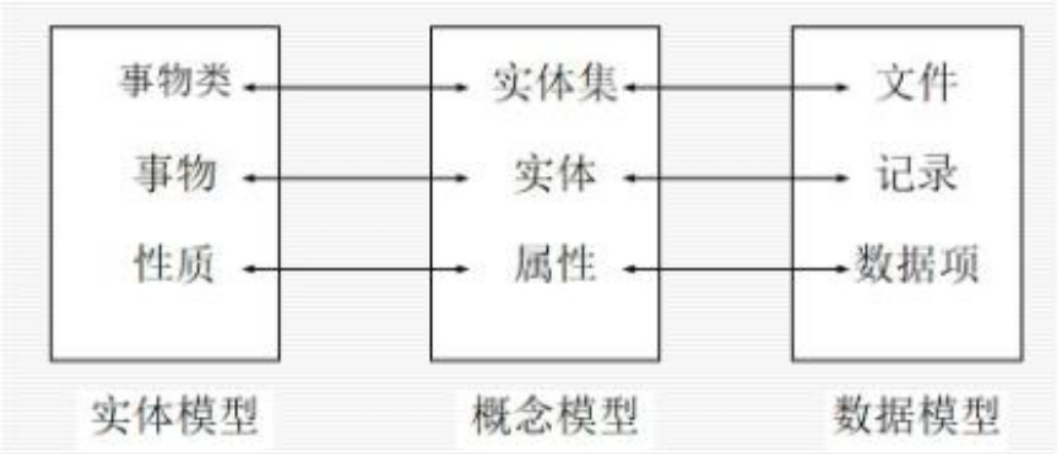
五、常见的关系型数据库管理系统

Oracle、SQL Server、Access、MySQL 等等。

六、上机操作：MySQL 的安装与配置（Windows 系统、Linux 系统）

登录数据库的方式（包括命令行下的用户名，密码；为了登录方便进行的环境变量的设置）。

七、数据库设计


**概念模型**：把现实世界转换为信息世界的模型，**E-R 模型**

**数据模型**：把信息世界转化为数据世界使用的模型，**关系模型**

query)、数据操纵(data manipulation)、数据定义(data definition)和数据控制（data control）功能于一体，充分体现了关系数据语言的特点和优点。

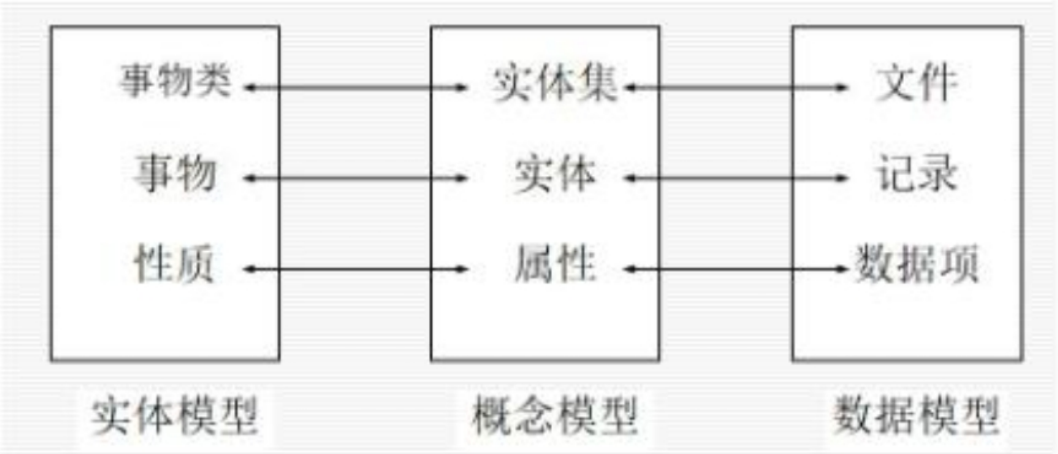
五、常见的关系型数据库管理系统

Oracle、SQL Server、Access、MySQL 等等。

六、上机操作：MySQL 的安装与配置（Windows 系统、Linux 系统）

登录数据库的方式（包括命令行下的用户名，密码；为了登录方便进行的环境变量的设置）。

七、数据库设计


**概念模型**：把现实世界转换为信息世界的模型，**E-R 模型**

**数据模型**：把信息世界转化为数据世界使用的模型，**关系模型**

标识对象（**实体**—Entity）

标识数据库要管理的关键对象或实体。

标识每个实体的**属性**（Attribute）

标识对象之间的**关系**（Relationship）

如何表示实体（什么图形）

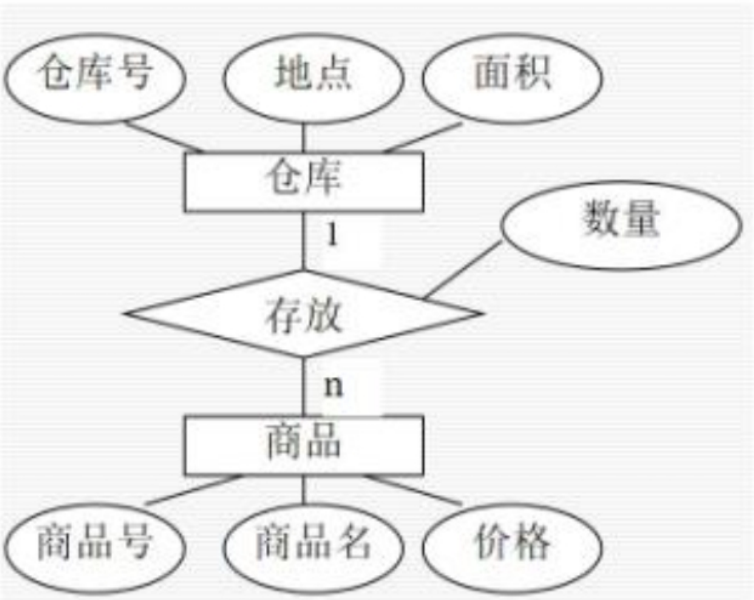
如何表示属性（什么图形）

**实体间不同联系情况**

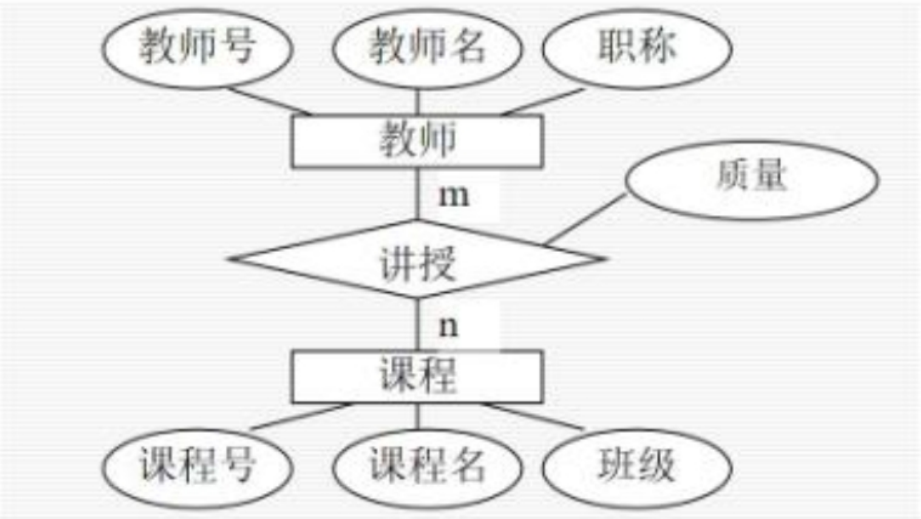
**一对一(1:1)联系**


主任与系之间的一对一的联系

**一对多（1：m）的联系**




**多对多（m:n)联系**




八、关系型数据库范式理论（略）

范式是具有最小冗余的表结构，Dr E.F.codd 最初定义了规范化的三个级别：

第一范式(1st NF －First Normal Fromate)

目标：确保每列的原子性

每列不可再分（也称为最小的原子单元）

第二范式(2nd NF—Second  Normal Fromate)

满足 1NF，且主键以外的其他列，都依赖该主键

每表只描述一件事情

第三范式(3rd NF— Third  Normal Fromate)

满足 2NF，且除主键外的其他列不传递依赖主键

## Chapter2 操作数据库和数据表

一、数据库的基本操作（具体操作实例详见课本）

**创建数据库**的语法为：

```
CREATE DATABASE database_name;
```

**删除数据库**语法为：

```
DROP DATABASE database_name;
```

**指定或切换数据库：**

```
USE <数据库名>
```

二、数据表的基本操作（具体操作实例详见课本）

**创建数据表：**

CREATE TABLE <表名>

(

列名 1，数据类型 [列级别约束条件] [默认值],

列名 2，数据类型 [列级别约束条件] [默认值],

.....

[表级别约束条件]

);

**注：数据表中字段的创建顺序与其在数据库中存储的顺序相同。**

理解数据库服务器、数据库、基本表以及后面视图的关系。

**主键**（重要），又称主码，是一列或多列的组合。主键约束（Primary Key constraint）要求**主键列的数据唯一，不允许为空。**

设置主键的方法：

单字段主键

**字段名 数据类型 PRIMARY KEY [默认值]**

多字段联合主键

**PRIMARY KEY [字段 1，字段 2,. . .，字段 n]**

**外键**用来在两表间建立链接，可以是一列或者多列。一个表可以有一个或多个外键。

创建外键的方法（略）：

[CONSTRAINT <外键名>] FOREIGN KEY 列名 1 [ ,列名 2,…]

REFERENCES <主表名> 主键列 1 [ ,主键列 2,…]

**保持数据引用的完整性，不允许删除具有关联关系的行。**

**注：主表（父表）：主键所在的表，班级表**

**从表（子表）：外键所在的表，学生表**

**非空约束（Not Null constraint）指字段的值不能为空。**

**字段名 数据类型 not null**

例：定义数据库表 tb_emp6，员工姓名不能为空

```
CREATE TABLE tb_emp6
(
id          INT(11) PRIMARY KEY,
name        VARCHAR(25) NOT NULL,
deptId      INT(11),
salary      FLOAT
);
```

**唯一性约束（Unique Constraint）要求该列唯一，确保一列或者几列不出现重复值。**

字段名 数据类型 UNIQUE

**默认约束**（Default Constraint）指定默认值。

字段名 数据类型 DEFAULT 默认值

**自动增加（例如，学号自动增加）**

AUTO_INCREMENT，特点。

以上约束条件中大部分属于单表约束，但外键约束需要用到两个关联表所以属于多表之间的约束。

**查看表基本结构语句 DESCRIBE**

DESCRIBE <表名>/DESC <表名>

**查看表详细结构语句 SHOW CREATE TABLE**

SHOW CREATE TABLE <表名\G>

三、修改数据表

**修改表名**

ALTER TABLE <旧表名> RENAME [TO] <新表名>;

**把字段的数据类型转换成另一种数据类型。**

ALTER TABLE <表名> MODIFY <字段名> <数据类型>

**修改字段名**

ALTER TABLE <表名>

CHANGE <旧字段名><新字段名><新数据类型>;

**添加新的字段：**

ALTER TABLE <表名>

ADD <新字段名><数据类型>

[约束条件] [FIRST | AFTER 已存在字段名];

**将数据表中的某个字段从表中移除。**

ALTER TABLE <表名> DROP <字段名>;

四、删除数据表

**删除没有被关联的表**

DROP TABLE [IF EXISTS]表 1，表 2，．．．表 n;

**删除被其它表关联的主表**

先删除关联的子表，再删除父表。

要保留子表，需将关联的表的外键约束取消，然后删父表。

**注：关联的数据表不能直接删除。**

Delete from 表名；　（因为没有指定删除记录的条件，所以清空表中的所有记录）

## Chapter3 数据类型和运算符

概述：

**数值型：**

整数型：TINYINT、SMALLINT、MEDIUMINT、INT、BIGINT

浮点型：FLOAT 和 DOUBLE

定点小数：DECIMAL

**日期/时间：**YEAR、TIME、DATE、DATETIME 和 TIMESTAMP

**字符串：**CHAR、VARCHAR、BINARY、VARBINARY、BLOB、TEXT、ENUM 和 SET 等。

### 一、整数类型

| 类型名称 | 说明 | 存储需求 | 有符号 | 无符号 |
|---|---|---|---|---|
| TINYINT | 很小的整数 | 1字节 | -128~127 | 0~255 |
| SMALLINT | 小的整数 | 2字节 | 32768~32767 | 0~65535 |
| MEDIUMINT | 中等大小的整数 | 3字节 | 8388608~8388607 | 0~16777215 |
| INT(INTEGER) | 普通大小的整数 | 4字节 | | |
| BIGINT | 大整数 | 8字节 | | |

**主要 int，用的最多。**

注：

浮点数类型和定点数类型

（1）显示宽度和取值范围无关，指最大可能显示的数字个数，小于

指定宽度时由空格填充。

（2）如果大于显示宽度，只要不超过该类型的取值范围，可以插入并显现。例如：year=19999。

二、浮点数类型和定点数类型

浮点数和定点数表示小数。

浮点类型：FLOAT、DOUBLE。

定点类型：DECIMAL。

**都可用(M,N)表示，M为精度，总位数；N为标度，小数位数。**

精度要求高，如money，用decimal，float、double易产生误差

三、日期与时间类型

| 类型名称 | 日期格式 | 日期范围 |
| --- | --- | --- |
| YEAR | YYYY | 1901~2155 |
| TIME | HH:MM:SS | -838:59:59~838:59:59 |
| DATE | YYYY-MM-DD | 1000-01-01~9999-12-3 |
| DATETIME | YYYY-MM-DD HH:MM:SS | 1000-01-01 00:00:00~9999-12-31 23:59:59 |
| TIMESTAMP | YYYY-MM-DD HH:MM:SS | 1970-01-01 00:00:01 UTC~2038-01-19 03:14:07 UTC |

**注：YEAR，TIME，DATE，DATETIME，TIMESTAMP五种数据类型的注意细节和示例再看下课本。**

四、**字符串类型**

存储字符串数据，如图片和声音的二进制数据。

CHAR、VARCHAR、BINARY、VARBINARY

BLOB、TEXT、ENUM 和 SET。

注：看下 CHAR、VARCHAR、BINARY 、BLOB、TEXT、ENUM，其它可略看。

五、二进制类型（略）

存储二进制数据的数据类型

BIT、BINARY、VARBINARY、TINYBLOB、BLOB、MEDIUMBLOB、LONGBLOB

六、常见运算符介绍

算术运算符比较熟悉。

比较运算符

比较运算符的结果是 1,0 或者是 NULL。包括：

=、<=>、<> (!=)、<=、>=、>、IS NULL

IS NOT NULL、LEAST、GREATEST、

BETWEEN . . . AND. . . 、ISNULL、IN、NOT IN、LIKE（可以使用的通配符只有%和_两个）、REGEXP

注：**(1) 运算优先级**

**(2) 看下比较陌生的几个：比如 BETWEEN . . . AND. . 、 NOT**

IN、LIKE 以及 REGEXP，尤其注意比如 BETWEEN . . . AND. .比较时的边界值（即等于的时候算不算）

逻辑运算符

逻辑运算符的求值所得结果为 TRUE、FALSE 或 NULL。

NOT 或者 !

AND 或者 &&

OR 或者 ||

XOR

注：看下书上的示例，理解一下运算关系。

## Chapter4 插入、更新与删除数据

### 一、插入数据（重点，示例详见课本）

```
INSERT INTO table_name(column_name)
  VALUES(value_list);
```

注：（1）插入数据时，保证值顺序与列字段的顺序相同；

（2）插入数据时允许名称列表为空，此时顺序和字段定义时的顺序相同

为指定字段插入数据，其他字段的值为默认值。

**【例2】在person表中，插入新记录，name值为Willam，age值为20：**

```
INSERT INTO person (name, age)
VALUES('Willam', 20);
```

**注：如果某个字段在定义时添加了非空约束，但没有添加 default（默认）约束，那么插入新记录时就必须为该字段赋值，否则数据库系统会提示错误。**

可以一条语句中插入多条记录的数据值。

**更新数据（重要）**

```
UPDATE table_name
SET column_name1 = value1,...,column_namen = valuen
WHERE (condition);
```

**【例5】更新id为11的记录，将age值改为15，将name值改为LiMing：**

```
UPDATE person SET age = 15, name='LiMing'
WHERE id = 11;
```

**删除数据（重要，注意和删除表的区别）**

```
DELETE FROM table_name [WHERE (condition)];
```

【例6】person表中，删除id等于11的记录：

```
DELETE FROM person WHERE id = 11;
```

**【例7】删除person表中所有记录：**

```
DELETE FROM person;
```

注：重点复习增删查改的命令行

Chapter5　索引

5.1 索引简介

1.什么是索引

索引是对数据库表的一列或多列的值进行排序的一种结构，使用索引可提高数据库中特定数据的查询速度。

索引是一个单独的、存储在磁盘上的数据结构，包含着对数据表中所有记录的引用指针。

2.索引的特点

索引的优点：

（1）通过创建唯一索引，可以保证数据库表中每一行数据的唯一性。

（2）可以大大地加快数据的**查询速度**，这是创建索引**最主要的原因。**

（3）在实现数据参考完整性方面，可以加速表和表之间的连接。

（4）在分组和排序时，可以显著减少分组和排序的时间。

索引的缺点：

（1）创建和维护索引要耗费时间，随着数据量的增加，所耗时间会

增加。

（2）索引需要占磁盘空间，如果有大量索引，索引文件就可能比数据文件更快达到最大文件尺寸。

（3）对表中的数据进行增删改时，索引也要维护，降低了数据的维护速度。

3.索引的分类

（1）普通索引和唯一索引

（2）单列索引和组合索引

（3）全文索引

（4）空间索引

4.索引的设计原则

（1)索引并非越多越好。不仅占用磁盘空间，而且影响INSERT、UPDATE、DELETE等语句的性能，因为表中的数据更改的同时，索引也会进行调整和更新。

（2）避免对经常更新的表进行过多的索引，并且索引中的列尽可能少。经常用于查询的字段应该创建索引，但要避免添加不必要的字段。

（3）数据量小的表最好不要使用索引。查询花费的时间可能比遍历索引的时间还要短，索引可能不会产生优化效果。

（4）在不同值较多的列上建立检索，在不同值少的列上不要建立索引。比如在学生表的“性别”字段上只有“男”与“女”两个不同值，因此无需建立索引。如果建立索引，会严重降低数据库的更新速度。

（5）当唯一性是某种数据本身的特征时，指定唯一索引。确保定义

的列的数据完整性，以提高查询速度。

（6）在频繁排序或分组（即进行 group by 或 order by 操作）的列上建立索引。如果待排序的列有多个，可以在这些列上建立组合索引。

5.多种方法创建索引

（1）CREATE TABLE 指定索引列；

（2）ALTER TABLE 在存在的表上建索引；

（3）CREATE INDEX 在存在的表上加索引。

6.代码

（1）创建索引：看一个示例会用即可。

比如：P122 例 5.1，或 P130 例 5.10，或 P132 例 5.13.

（2）删除索引：看一个示例会用即可。

比如：P134 例 5.19.

Chapter6 视图

1.什么是视图

视图是从一个或多个表中导出的，视图的行为与表非常相似，但视图是一个**虚拟**表。

在视图中，用户可以使用 SELECT 语句查询数据，以及使用 INSERT、UPDATE 和 DELETE 语句修改记录。

视图可以使用户操作方便。

**保障数据库系统的安全。**

2.视图的含义

通过视图看到的数据只是存放在基本表中的数据。

**对通过视图数据修改，相应的基本表的数据也要变化；**

同时，**基本表的数据变化，则自动反映到视图中。**

注：视图不存储数据。

视图和基本表的区别和联系。

3.视图的作用

（1）简单性

被经常使用的查询可以被定义为视图，不必每次指定全部的条件。

（2）安全性

通过视图，用户只能查询和修改所见到的数据。

数据库授权命令可以限制到特定的数据库对象，但不能授权到数据库特定的行或列，通过视图，可以解决。

（3）逻辑数据独立性

帮助用户屏蔽真实表结构变化带来的影响。

4.视图的操作

创建视图代码：可参考一到两个示例。

比如：P148 例 6.1，或例 6.2.

修改、更新和删除(drop)视图的代码，每个参考一个示例。P154-P158（不用看语法，只看示例即可）。

注：视图修改是指对视图结构的修改，比如字段的增删等。

视图更新是指对视图数据的更新，即对行上的一条条的记录。

注意对比：视图数据更新，基本表的数据更新了没有；

基本表数据更新，视图的数据更新了没有。（P155-P158给了答案）

Chapter7　　重点复习命令行，参考 chapt7（7.2.8）.txt。