

Introduction to Database (H)

Yanqiao Chen, SUSTech, Shenzhen

Instructor: Shiqi Yu

September 19, 2025

Contents

| | |
|---------------------|----------|
| 第一章 数据库引论 | 5 |
| 1.1 数据库历史 | 5 |
| 1.1.1 为什么需要数据库? | 5 |
| 1.1.2 关系数据库 | 5 |
| 1.1.3 范式 | 6 |
| 第二章 SQL 语句 | 7 |
| 2.1 C/S 结构 | 7 |
| 2.2 SQL 语句标准 | 7 |
| 2.3 DDL 语句 | 7 |
| 2.3.1 CREATE | 7 |
| 2.3.2 Null | 8 |
| 2.3.3 ALTER | 9 |
| 2.3.4 DROP | 9 |
| 2.3.5 INSERT | 9 |
| 2.4 DML 语句 | 9 |
| 2.4.1 SELECT: 查询 | 9 |
| 2.4.2 UPDATE | 10 |
| 2.4.3 DELETE | 10 |
| 2.5 Schema | 10 |
| 2.6 Window Function | 10 |
| 2.7 Trigger | 10 |
| 2.8 一些其它语句 | 10 |
| 2.8.1 not in | 10 |

第一章 数据库引论

1.1 数据库历史

1.1.1 为什么需要数据库？

我们需要管理数据，查询、修改数据，同时需要管理多个用户读写的并发问题，因此需要开发一套合适的数据库系统进行数据管理。

1.1.2 关系数据库

Edgar Codd 在 1970 年代提出的关系模型，为后续关系数据库的开发奠定了基础。关系数据库的最小单位是“行”，再上面还有页（Page）、表（Table）。关系数据库中存在很多操作，一些基本操作是：

- $\text{select } (\sigma_P(\text{tag}))$ ：取出符合输入关系满足谓词的行；
- $\text{project } (\Pi_{\text{tag1}, \text{tag2}}(\text{tag}))$ ：对输入关系所有的行输出指定的属性。从输出中取出重复元组。
- join ：输出两个输入关系中元组的并。
- product ：从两个输入关系中输出所有元组对；
- concat ：从两个输入关系输出这样的元组对：它们在具有相同名字的所有属性上取值相同。

数据库中，每一行本质上是没有顺序的，也就是一个集合（无序、不重合）。为了区分每一个数据，我们为每一个数据定义一个键值（Key）：

Definition 1.1 (键值). 键值是识别一条数据的唯一表示，可以是对于某些数据的哈希值，或者是按照初次写入顺序的标号。总而言之，键值必须要是独一无二的。

Remark 1.1 (Primary Key). 我们可能使用很多键值，但是我们可以选出一个基本的键值作为“*Primary Key*”。一般来说，我们使用整数来进行标号。

对于很多数据的写法，我们需要对数据进行规范化，方便进行查询和修改。

Definition 1.2 (规范化). 对于指示同一个属性的标签, 标签可能有很多种写法。同样, 数据的标签可能根本不存在。我们在设计相关的标签的时候要考虑到这一点。

例如, 汉族人通常有姓和名, 但是其它民族的人可能没有姓氏, 外国人可能存在姓名倒置, 可能存在昵称等等。这里, 我们需要设计一些机制对标签进行规范化。

Example 1.1. 例如, 可以将另一个表的索引作为标签, 从而化简一张表上的数据储存。

Example 1.2. 还可以, 我们将不同类型的数据分开存, 通过第三张表记录它们的关系。这就是关系数据库的基本思想。

表的关系可以视为一张无向图。

1.1.3 范式

在设计数据库的过程中, 我们往往遵循一些工程范式。

- Simple Attribute (属性原子化);
- Attribute depend on Full Key (属性描述键);
- Non-Key Attribute do not depend on each other (属性不相关);

第二章 SQL 语句

SQL 语言在不同的数据库中都会有些许不同，这被称为 SQL 方言。因此，我们在使用对应的数据库的时候，都需要进行一定程度上的 DIY。SQL 是大小写不敏感的，但是尽量使用大写标识关键字，小写标识标识符（注意，数据大小写有别）。

2.1 C/S 结构

我们所安装的 Postgre 是一个数据库服务器 (Server)，我们需要通过客户端 (Client) 访问服务器。首先，我们通过服务器与服务器端口建立连接，通过用户名和密码登入服务器。然后，我们通过客户端的图形界面输入 SQL 语言与服务器进行交互。我们的执行顺序如下：

- 客户端输入 SQL 查询（或者数据处理）命令
- 服务器接受 SQL 命令，挂起事务，检查权限，处理数据，发回客户端；
- 客户端接收服务器的数据；

2.2 SQL 语句标准

每一个 SQL 语句由一个命令序列，一个命令是一个记号序列（记号可以是一个关键字、一个标识符、一个带引号的标识符、一个 literal 或者常量、一个特殊字符）并且由一个分号终结。

2.3 DDL 语句

2.3.1 CREATE

CREATE 语句用于创建表、行等等。在创建表的过程中，每一列的属性具有特定的类型，例如整数、字符串等等。SQL 的数据类型包含

- **Text:** `char(length)` , `char` 默认长度为 1, `varchar`（可变长度，不超过最长长度）, `clob`, `varchar2`(Oracle 方言);

- **Number:** `int`, `float` (浮点数的数字标识范围大于其位宽, 易出错), `numeric`(精确位, 小数位) (精确数值类型);
- **Date:** `date` (Oracle 方言), `datetime`, `timestamp`;
- **Binary:** `raw(max length)`, `varbinary(max length)`, `blob`, `bytea`.

CREATE TABLE 语句的基本形式如下:

```
1 CREATE TABLE 表名(  
2     标识符 类型 (not null, 如果不能留空) (primary key, 隐式not null, 不重  
3     复, 只有一个),  
4     check(条件) ,  
5     标识符 类型 ,  
6     .....  
7     unique(标识符, 标识符)  
);
```

Constraint 创建表的时候, 我们可以添加一系列限制来规范输入:

- **not null:** 附加在标识符之后, 表示填入数据列不得为 `null`;
- **primary key:** 附加在标识符后, 隐含 `not null` 和 `unique`; 单独成行可以标记基本键值元组;
- **check:** 添加在标识符后方, 多个用逗号分隔: `check(bool expr)`;
- **unique:** 用逗号分隔, 可以限制单个元素唯一, 也可以限制某一个元组唯一; 或者, 直接加在标识符后方;
- **constraint:** 若独立成行, 可以标记外键 (`constraint 限制名 foreign key(外键名) references 表名 (键名)`); 总之是用来命名“限制”的;
- **foreign key:** 标准语法是: `foreign key 键或者元组 references 表名 (键或者元组)`;
- **default:** 附加在键后面, 表示若没有显式初始化, 则直接初始化为此值;
- **references:** 附加在键后面可以表示参考的外部键值是什么, 保持内外一致;

2.3.2 Null

`Null` 指的是“不存在值”, 与其它编程语言不同, 不能够运算。

2.3.3 ALTER

2.3.4 DROP

2.3.5 INSERT

2.4 DML 语句

2.4.1 SELECT: 查询

Select 语句的语句结构如下:

```
1  [WITH [RECURSIVE] with_query [, ...]]
2  SELECT [ALL | DISTINCT [ON (expression [, ...])]]
3      * | expression [ [AS] output_name ] [, ...]
4  [FROM from_item [, ...]]
5  [WHERE condition]
6  [GROUP BY grouping_element [, ...]]
7  [HAVING condition]
8  [WINDOW window_name AS (window_definition) [, ...]]
9  [{UNION | INTERSECT | EXCEPT} [ALL | DISTINCT] select]
10 [ORDER BY expression [ASC | DESC | USING operator] [NULLS {FIRST | LAST}]
    [, ...]]
11 [LIMIT {count | ALL}]
12 [OFFSET start [ROW | ROWS]]
13 [FETCH {FIRST | NEXT} [count] {ROW | ROWS} ONLY]
14 [FOR {UPDATE | NO KEY UPDATE | SHARE | KEY SHARE} [OF table_name [, ...]]
    [NOWAIT | SKIP LOCKED] [...]]
```

2.4.2 UPDATE

2.4.3 DELETE

2.5 Schema

2.6 Window Function

2.7 Trigger

2.8 一些其它语句

2.8.1 not in