**Week1**

**DML**: Data Definition Language

* Select, insert, update, delete

**DDL**: Data Definition Language

* Create, Alter, Drop

**DCL**: Data Control Language

* Grant, Revoke

**TCL**：控制事务。

COMMIT、ROLLBACK、SAVEPOINT

**Logical data independence**:

protection of the applications from changes in the logical structure of the data

**Physical data independence**:

protection of the conceptual schema (and applications) from physical layout changes

**Week2**

* 需要能够甄别出ERD中的错误
* 能根据plain text画出对应的ERD，重点关注Constraint部分
* 箭头从Entity指向Relationship

**Entity**

Rectangle

**Weak Entity** (1:N relationship -> Strong Entity 和 关系之间不能有箭头，否则破坏了1:N关系)

Double Diamond – weak relationship

Double Rectangle – Weak Entity

Dotted Underline – Discriminator (Partial Key), weak entity must use discriminator and strong entity’s PK to Form a Composite PK

**Attributes**

Eclipse

**Multi-value Attributes**

Double Eclipse

**Composite Attributes**

多画一条线连起来

**Derived Attribute**

Dot Eclipse

**Primary Key**

Underline

**Foreign Key**

Does not show

**Partial Key**

Doted Underline

**Relationship** (不需要画Key, 因为用连接的Entity PKs作为组合主键)

Diamond

**Aggregation Relationship**

Dotted Box

Conveys a relationship between two relationships

**Key Constraint (N-to-1)**

Thin Arrow

At most one

**Total/Partial Participation (N-to-M)**

Total: Thick Line; at least one

Partial: Thin Line

**Combine Key and Total Participation**

Thick Arrow

Exactly One

**Cardinality**

用label在线旁边表示

\* 表示无限大

0…1 用细线

1...3 用粗线

**IsA**

这里没有thick arrow（即一个父类必须属于一个子类）, 但是等同于thick line + Disjoint

没有 cardinality

**Overlap Constraint**

Disjoint: 只能属于最多一种子类

标注在线旁边

Overlapping (default): 可以属于多个子类

无需表示

**Covering Constraint**

Total: 父类必须属于子类

Thick Line

Partial (default): 父类可以不属于子类

Thin Line

**Week3**

* 知道RM如何定义FK constraint才能满足ERD, 比如FK加上NOT NULL就能表示exactly one

**Real word Relation — RDBMS**

* allow duplicated rows
* support ordering tuples and attributes
* allows “null”

**Advantage/Disadvantage of NULL**

Advantage:

* NULL can be useful because using an ordinary value with a specific meaning may not always work. 比如求mean的时候，如果我们用-1代表null就会出错

Disadvantage:

* NULL may cause complications in the definition of many operations

**CHAR VS VARCHAR**

CHAR 是固定长度

VARCHAR 是动态长度

**CREATE TABLE NAME(…);**

**DROP TABLE NAME CASCADE**

**ALTER TABLE**

* **添加列**: ALTER TABLE Flight ADD test1 INTEGER, ADD test2 INTEGER;
* **删除列**: ALTER TABLE Flight DROP test1;
* **添加Constraint**: 只能先删除旧的外键约束，再添加一个新的外键约束
  + ALTER TABLE Orders ADD FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id);
  + ALTER TABLE Orders ADD UNIQUE (column\_name);
* 删除Constraint: **除了NOT NULL以外，别的都需要对应的Constraint名字来删除**
  + ALTER TABLE table\_name DROP CONSTRAINT column\_name;
  + ALTER TABLE table\_name ALTER COLUMN column\_name DROP NOT NULL;
* **重命名列**: ALTER TABLE table\_name RENAME COLUMN old\_column\_name TO new\_column\_name;
* **重命名表**: ALTER TABLE old\_table\_name RENAME TO new\_table\_name;
* 修改Default Value 或者 data type
  + ALTER TABLE table\_name ALTER COLUMN column\_name SET DEFAULT 7.77;
  + ALTER TABLE table\_name ALTER COLUMN column\_name TYPE new\_data\_type;

**PRIMARY KEY（主键）**

* **唯一性**：每个表只能有一个主键（At most one per table），并且主键字段的值必须是唯一的。
* **不允许NULL值**：主键字段不能包含空值（Automatically disallows NULL values）
* **Composite PK可以自己设定**，但必须满足唯一性约束。

**CANDIDATE KEY**

* **唯一性**：候选键中的字段也必须是唯一的（all must be declared as UNIQUE）。
* **最小性（Minimality）**：没有多余的属性，可以删除任何一个属性而不再满足唯一性的条件。这里的最小是不是指size，而是说不能去掉任何属性。
* **可以包含NULL值**

**SUPER KEY**

* **如果只满足唯一性，那么就叫做Super Key**
* **所有Candidate key 和 PK 都是 Super key**

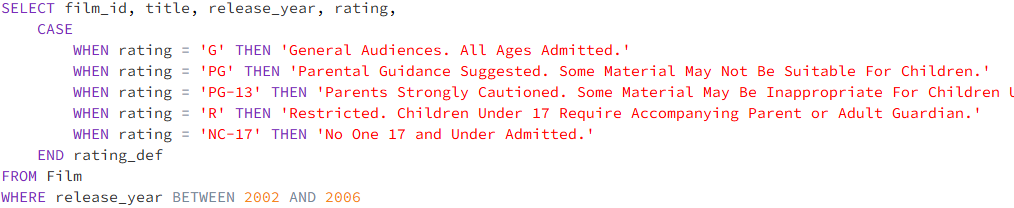
**Mapping relationship types From ERD To RM**

* 不包含derived variable
* FK的箭头应该指向被引用的主键
* 实线表示PK, 虚线表示FK
* 可以有多条实线组成Composite PK
* 一个attribute可以同时有虚线和实线，表示它同时属于composite key和foreign key，通常出现在Weak Entity和 IsA relationship中
* 在IsA relationship中，子类应该和父类共享Primary Key

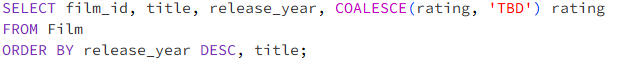
**Week4**

Attention in SQL syntax

* 英文
  + chronologically 按时间顺序
  + ordinal ranking: 每个条目都有独立编号，即使有相同值，也不会并列；ROW\_NUMBER()
  + dense ranking：如果有相同值，会并列排名，下一个排名不会跳过；DENSE\_RANK()
  + Standard ranking（或叫 sparse ranking）：有相同值时会并列，下一个排名会跳过；RANK()
* REFERENCE KEY MUST EITHER BE CANDIDATE KEY OR UNIQUE
* 不是每个表都必须有主键，但是绝大部分正规设计的表都会有主键。
* BETWEEN .. AND... 是inclusive的，而且可以对String使用
* Int / Int 返回的还是INT，并且是直接截断小数部位，不会四舍五入
* LIKE 和 LOWER连用可以解决忽略大小写的问题
* NULL 用 IS 判断
* 需要注意的是，别名需要用双引号 或者 不带引号，不能用单引号
* 用YEAR作为别名的时候要用到AS
* 关于SELECT中别名的引用
  + WHERE 里面不能用
  + HAVING 里面不能用
  + GROUP BY 可以
* natural join 不用关心表的顺序，它会自动找出其中相匹配的col进行连接，并且会合并同名列
* The set operators require that all set relations have the /same schema/.
* INFINITY 和 -INFINITY 和 NaN 都是FLOAT 类型，并且INFINITY = INFINITY
* TIME, DATE, TIMESTAMP
  + CURRENT\_TIMESTAMP, CURRENT\_DATE, CURRENT\_TIME
  + 可以用EXTRACT提取对应的,比如TIME能提取MINUTE，但是DATE不行
    - DAY
    - MONTH
    - YEAR
    - HOUR
    - MINUTE
    - SECOND
  + DOY (day-of-the-year)
  + TIMEZONE, TIMEZONE\_HOUR, TIMEZONE\_MINUTE
    - SELECT EXTRACT(**TIMEZONE** FROM TIMESTAMP '2025-05-04 12:00:00+08:00'); 像这里是28800 秒 对应的是 8 小时
    - SELECT EXTRACT(TIMEZONE\_**HOUR** FROM TIMESTAMP '2025-05-04 12:00:00-08:00'); -8 因为小时部分为-8
    - SELECT EXTRACT(TIMEZONE\_**MINUTE** FROM TIMESTAMP '2025-05-04 12:00:00+08:00'); 0 因为分钟部分为0
  + INTERVAL
    - INTERVAL '5 days'
  + DATE – DATE 得到的是相差的天数，为INT
  + DATE – INTERVAL 得到的还是 DATE类型
* Format String
  + TO\_CHAR(5, 'FM00') 会返回 05，即使数字是单一数字，FM 也会强制将它变成两位数字。
  + TO\_CHAR(5, 'FM00x') 会返回 05x，因为x并不会被识别成填充符号。
  + TO\_CHAR(-5, 'S00') 会返回 -05
* CASE WHEN … THEN … ELSE … END



* COALESCE()



* NULLIF()

A close up of text

AI-generated content may be incorrect.

* STRING\_AGG(expression, separator ORDER BY attributes) 他就是一个聚合函数，和MAX()之类的并无差别

Example





A black and red text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

也就是说在这里，如果用length/3，那么当length = 181的时候就不会算入，因为 181/3=60

A close-up of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

Write an SQL query that finds all films (by film\_id and title) categorised as both “Drama” and “Family” film.

A screenshot of a computer program

AI-generated content may be incorrect.

在Cartesian里面，自己是会和自己生成row的

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of text

AI-generated content may be incorrect. A screen shot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

可以在select和where里面使用

ROUND()是四舍五入, 对于 150/60 来说, 结果为2; 但是对于ROUND(150/60,0)来说就是3, 对于ROUND(150/60)来说就是2

 A close up of a grid

AI-generated content may be incorrect.

如果不想四舍五入就用TRUNC()

A screen shot of a computer code

AI-generated content may be incorrect.

A close up of numbers

AI-generated content may be incorrect.

A close up of a message

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A close up of text

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

A computer code with text

AI-generated content may be incorrect.

A screen shot of a computer code

AI-generated content may be incorrect.