

E-R图

方块 : entity , 菱形 : relationship , 椭圆 : attribute

studnet<-(0,M)-----takes----->(1,M)->courses : student可以take1到M节课, 反过来一节课可以有0到M个学生

0:0 ; 1 : | ; many : <

没有intersect只有innerjoin

底层逻辑

执行顺序 : from->where->group by->having->select->order by

在sql standard里面, not and or没有计算优先级, 从左到右, 但是具体的oracle, MongoDB可能会有。建议使用括号

select 里面的聚合函数, 在where里面可以使用列名代替

语法

%代表任意长度 (0-无穷) 的字符 ; _代表单个字符 ; %也指除余

left(201819,4)->2018 ; right(201819,4)->1819 ; substring(abcde,2或者-4,2)->bc ; -substring(abcde,3)->cde

union ; union all不去重 ; intersect ; except

#sub-queries : 子查询的order by不生效, 在比较中只能出现在右侧, 不能用作非比较的表达式 (比如我在select里定义了一个子查询, 在group by里是不能用的)

non-correlated : 为外循环执行一次

correlated : 利用外循环的数据执行内循环, 为外循环执行很多次, 每次用一条数据, 可以用exists

exists 的return值是True/False ; 如果子查询的结果非空则True, 空则False

语法 (cont)

集合运算subquery :

把子查询当成一张表来看 : 下面从两张表里选择, where里比较的两列也出自两张表

```
SELECT xxx FROM (SELECT AVG(xx)
AVGPRICE FROM PRODUCT_T) AS
AVGPRICE_T, PRODUCT_T WHERE
STANDARD_PRICE > AVGPRICE
```

把子查询的结果当成一个数字来看 : select, where里面插子查询

```
SELECT xxx, salary - (SELECT AVG(salary)
FROM Staff) As SalDiff FROM Staff
WHERE salary > (SELECT AVG(salary)
FROM Staff)
```

相关子查询

Select * from books as a Where 价格> (select avg(价格) from books as b where a.类编号=b.类编号)

all();any() ; some() : 跟exists类似

重复一个数字times次 : rep(num,times)

case when then else end , 默认else是null

(Employee.DepartmentId , Salary) in (select XXX)

时间函数

字符串-日期 : str_to_date('01-Jan-202-3','%d-%b-%Y')

日期-时间 : date_format(date,'%Y-%m-%d')

create语句

create schema

create table db.tablename (

colname, datatype, not null/ default null/ auto increment.....,

primary key (colname)

constraint "keyname"

create语句 (cont)

foreign key ("product id")

references db.table ("colname")

on delete no action (父母表报错, 删除指令被roll back) /cascade (父母表的删掉, 子表的也删掉) /set null, default

on update(same as delete)

constraint check ("colname" in (a,b,c,d,e))

create view

改

#改表属性

alter table tablename

add colname datatype

DROP CONSTRAINT StaffNotHandlingTooMuch

#改表内容

UPDATE table SET colname = 775

WHERE xxx

增

INSERT INTO CUSTOMER_T VALUES

(001, 'Contemporary Casuals', '1355 S. Himes Blvd.', 'Gainesville', 'FL', 32601)

#有null的时候使用

INSERT INTO PRODUCT_T (PRODUCT_ID, PRODUCT_DESCRIPTION, PRODUCT_FINISH, STANDARD_PRICE, PRODUCT_ON_HAND) VALUES

(1, 'End Table', 'Cherry', 175, 8)

#从其他表里选取插入 INSERT INTO CA_CUSTOMER_T SELECT * FROM CUSTOMER_T WHERE STATE = 'CA'

#有auto increment的列不必自己定一个值

insert into db.table (colname) values ('Ben'), ('Jayden')



By cgeeeeh

cheatography.com/cgeeeeh/

Not published yet.

Last updated 24th September, 2023.

Page 1 of 2.

Sponsored by [ApolloPad.com](https://apollopod.com)

Everyone has a novel in them. Finish Yours!

<https://apollopod.com>

删

delete from db.table where xxx

delete from table

连接

join : 卡式乘积

natural join : t1 natural join t2。找t1的主键, t2的外键, join在一起, 消灭重复的列

join的时候可以用=, 也可以用>, <

多表连接: 可以用from+多表+where 多个=, 相当于inner join, 不在乎连接的顺序, 不像left join

```
FROM CUSTOMER_T, ORDER_T,  
ORDER_LINE_T, PRODUCT_T
```

```
WHERE CUSTOMER_T.CUSTOMER_ID =  
ORDER_T.CUSTOMER_ID AND ORDER_  
T.ORDER_ID = ORDER_LINE_T.OR  
DER_ID AND ORDER_LINE_T.PRODU  
CT_ID = PRODUCT_T.PRODUCT_ID AND  
ORDER_T.ORDER_ID = 5105
```

view

```
CREATE VIEW view_name (col1, col2,  
col3) AS select xxx
```

```
DROP VIEW ViewName [RESTRICT |  
CASCADE]/ restrict是删除, 如果有其他v-  
iew依赖这个view, 删除会被reject. casc-  
ade会连带着把依赖的view都删掉
```

Full-text Search

```
WHERE MATCH(Question_Answered)  
AGAINST("why, invention")
```

窗口函数

窗口函数不能跟groupby一起用, 最好别放进where, 不能用计算得出的列做新的判断

但是可以在select里面用来计算新的值

```
rank() over(partition by order by)
```

row_number是unsigned, 做减法出现负数的时候会报错, cast(as signed)即可

刷题经验

1、没有计算结果也需要的, 往往用join完成null, 再把null转成数值

2、case when then else end 和if, ifnull等可以用来转换一列数据, 比如apple->+, - orange -

3、限制行数的三种方法, limit 1, 窗口, 聚合函数

4、如果筛选条件有两个, 涉及到每个组内两个不同的数据部分。可以在having中用case when

5、如果要选取一列里的某些值, 而判断依据是group的结果, 只能用子查询

5、如果要不要选取一个值是由其他行的某个值决定的, 那也要用子查询

6、如果同一列自己要跟自己列的其他行比较, 用自联结

正则表达式

regexp, like

regexp : [1,2,3],[1-3],[a-z]。*代表0-m-any, .代表单个

like : %, _



By cgeeeeh

cheatography.com/cgeeeeh/

Not published yet.

Last updated 24th September, 2023.

Page 2 of 2.

Sponsored by [ApolloPad.com](https://apollopad.com)

Everyone has a novel in them. Finish

Yours!

<https://apollopad.com>