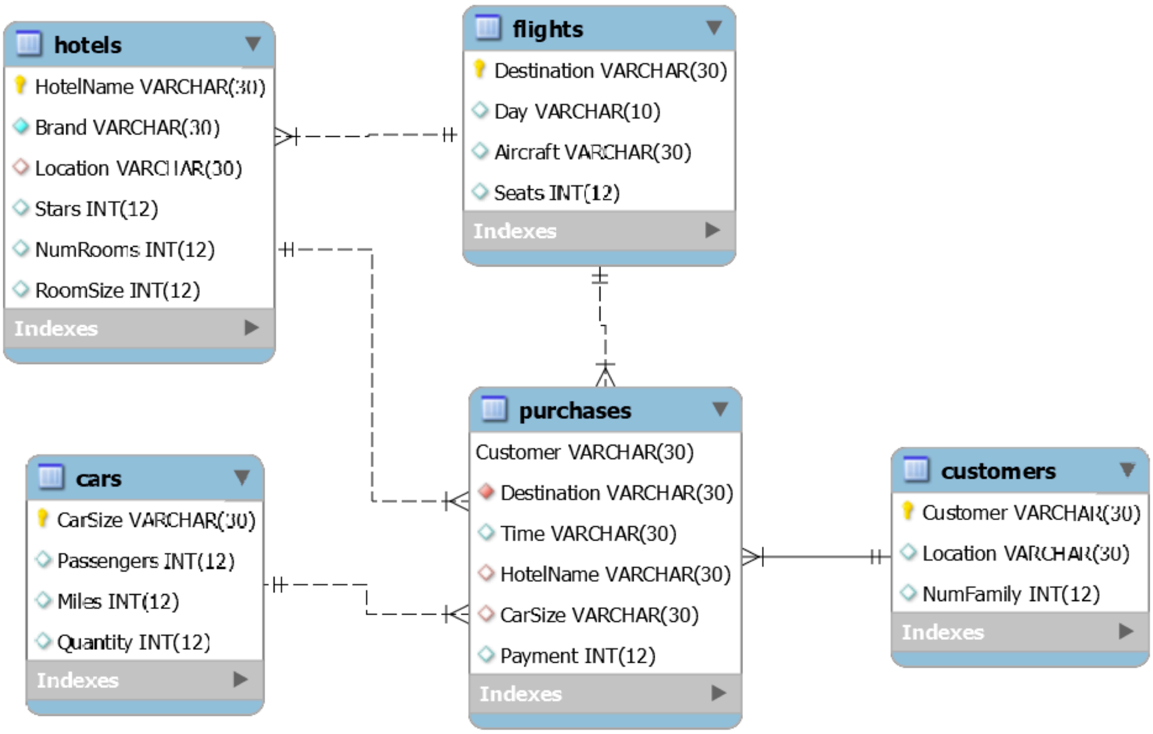
SQL CHEATSHEET



想好是哪个 JOIN（OUTER JOIN? ）

多个 where 语句注意重新写变量名

Repository AKA Data Dictionary

-Store metadata for the database

-include information on relationship between files or tables in a particular database

Cardinality of Relationships

Many-Many,

Many-One, arrow entering the one side

One-One, arrow entering both entity sets

Rounded Arrow: exactly one.

Multivalued Attributes

Double circle for the multivalued attributes

Keys-Primary and Foreign

Use foreign keys to relate relations

subclass

Use the isa to identify the subclasses

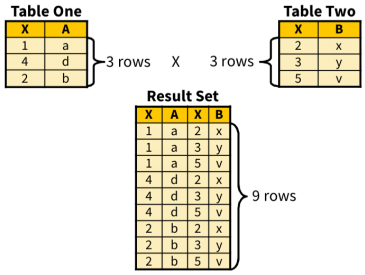
If there is overlapping between subclasses : O

No overlapping(Disjoint):d

Aggregations:

SUM AVG COUNT MIN MAX VARIANCE

用的时候注意：全部 aggregation 不然放在Group By



X	A
1	a
4	d
2	b

X	B
2	x
3	y
5	v

X	A	X	B
1	a	2	x
1	a	3	y
1	a	5	v
4	d	2	x
4	d	3	y
4	d	5	v
2	b	2	x
2	b	3	y
2	b	5	v

Null will be ignored in the aggregation

But count of the null is 0

JOIN RELATIONS

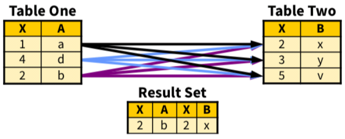
Cross Join(Cartesian Product)

select * from one, two;

INNER JOIN

select * from one inner join two on ...=....

or **using( ) 注意加括号!**



X	A
1	a
4	d
2	b

X	B
2	x
3	y
5	v

X	A	X	B
2	b	2	x

用 subquery 的时候一定要用 AS

Natural Join 不返回重复列

OUTER JOIN

#用 self join to find pairs

select distinct c1.CustomerName, c2.CustomerName, Address
FROM customers as c1 join customers as c2 using(Address)
where c1.customerName<c2.customerName
order by c1.customerName, c2.CustomerName, Address;

#用self join to select vendors and their maximum inventory.

select productName, productVendor, maxIn
from products right outer join(
select productVendor as pv, max(quantityInStock)as maxIn from products
group by productVendor) as p2 on products.productVendor = p2.pv
group by productVendor
order by productVendor;

#两层 join 写法 以及 where group by 顺序

```
select count(distinct customerName) as total_num, gender
from drivers join (customers join requests using(CustomerName))
using (driverName)
where color ='black'
group by gender;
```

IF 语句写法

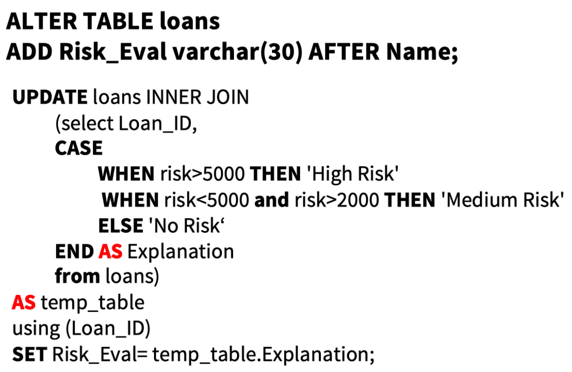
```
select customerName, count(RequestID), customers.pasttrips,
(count(requestID)+IF(Pasttrips is null,0,pasttrips)) as totaltrips
from requests right join customers using(customerName)
group by customerName
order by totaltrips LIMIT 12;
```

#Where 嵌套 subquery

```
select distinct customerName,age,destination
from requests join customers using (customerName)
where age>(select avg(age) from customers);
```

#添加删除列

```
alter table companies
add companySize Varchar(30) AFTER numemployees;
update companies join(select company,
case
        when NumEmployees>5000 then "large"
    else 'No information'
    end as CompanySize
from companies) as jointable
using (Company)
SET companies.CompanySize =
jointable.Companysize;
Where CompanySize is not '.....'
alter table companies
drop column CompanySize;
```



```
ALTER TABLE loans
ADD Risk_Eval varchar(30) AFTER Name;

UPDATE loans INNER JOIN
    (select Loan_ID,
    CASE
        WHEN risk>5000 THEN 'High Risk'
         WHEN risk<5000 and risk>2000 THEN 'Medium Risk'
        ELSE 'No Risk‘
    END AS Explanation
    from loans)
AS temp_table
using (Loan_ID)
SET Risk_Eval= temp_table.Explanation;
```

#用 all 语句找最大值

```
SELECT company, NumEmployees
from companies
WHERE NumEmployees>=all
(select NumEmployees from companies
where NumEmployees is not null);
```

#用 in 限制 where 语句,双重 query

```
select *
from companies where company in
(select company from attendance where careerfair = 'All Campus' and
company in
(select company from jobopenings where position = 'Data Scientist' or
position = 'Analyst'));
```

#双重 query 求均值，注意 having clause, 注意 subquery 选出来的数据集需要重新命名。

```
SELECT customerNumber, COUNT(orderNumber)
FROM orders
GROUP BY customerNumber
HAVING COUNT(orderNumber) >
(SELECT AVG(cOrders)
FROM(SELECT CustomerNumber, (COUNT(ordernumber)) AS cOrders
FROM Orders GROUP BY CustomerNumber) AS c1);
```

#用另外一个数据做排序

```
SELECT productcode, ordernumber, priceeach
from orderdetails right join
(select productCode from products order by productCode DESC LIMIT 20)
as TempProd USING(productCode)
order by productCode desc, orderNumber;
```

#Speical kind of self join

```
SELECT p1.productVendor, productCode, productName, quantityInStock,
MaxInventory
FROM Products AS p1 JOIN
(SELECT productVendor, MAX(QuantityInStock) AS MaxInventory
FROM Products GROUP BY productVendor )AS p2
ON p1.productVendor = p2.productVendor
WHERE p1.quantityInStock = p2.MaxInventory
```

# IF we need distinct productVendor

# As 要放在整个语句之后

```
select c1/c2 as turn, productVendor
from (
(select sum(quantityOrdered) as c1, productVendor
from orderdetails left join products using (productCode) group by
productVendor) as a
join
(select sum(quantityInStock) as c2, productVendor from products group by
productVendor) as b
using(productVendor) );
```

#Variable is not allowed in the exam

```
SET @var1:=
(select avg(quantityInStock) from products);
select productCode, quantityInStock from products
where quantityInStock < @var1
ORDER BY quantityInStock DESC;
```

#Create A Database

```
create database temp; use temp; create table bankcustomer
(contactID INT(11) primary key not null auto_increment, Name varchar(30)
not null, networth double)
insert into bankcustomer(contact......)
```

values ('','','',null) must include null value

CONCAT: set address = concat('name'-'bank')

#Patterns

```
select* from ** where names like 's%'
where names like 'S__' where names like '%m%'
```

ALL ANY 可以灵活运用在 where 和 having 语句

EXISTS 语句的应用

find customers who has not placed any orders

```
select customerNum, customerName from Customers
where not exists
(select * from orders where orders.customerNum=customers.customerNum)
```

LEFT JOIN 也可以解决类似问题