**面试的问题**

# oracle部分

#### 分区有哪些不同的类型？

范围分区，列表分区，散列分区，组合分区

#### 每种类型一般是用在什么场景下？

范围分区：用指定的分区键决定的范围进行分区，最为常用，分区键常采用日期。

列表分区：某列的值只有几个，可以采用列表分区。

散列分区：通过指定分区数量或编号来均匀分布数据的一种分区类型，分区数量常采用2的N次方；当列的值没有合适的条件时，采用散列分区；

组合分区:范围分区和列表分区组合；范围分区和散列分区组合，分区中的分区被称为子分区；

#### 哪种分区类型使用的最多？为什么？

看具体情况：

如果需要进行数据的过期化处理，那么范围分区基本上是唯一的选择。

如果需要数据的均匀分布，那么可以考虑使用HASH分区。

如果数据的值可以很好地对应于某个分区，那么就可以考虑使用列表分区。

在上面的原则基础上，再结合性能的影响因素，来最终确定使用哪种类型的分区。

#### 有没有用过组合分区？怎么搭配的？

范围+列表

范围+散列

列表+散列

#### 分区表创建的基本语法是什么？

Create table 表名

（字段名，字段类型……）

partition by range(字段名)

（partition 子分区名 values less than (某个日期) ）

#### 分区还有哪些不同的操作？

添加分区，删除分区，截断分区，合并分区，拆分分区，重命名分区

#### 怎么去查看表格分区的信息？

select \* from USER\_TAB\_PARTITIONS a where a.table\_name = '表名';

#### 对分区字段进行where筛选和直接读取分区内容，谁快一些？

select \* from fq\_emp\_list where depnto=10;

**select \* from fq\_emp\_list partition(d10); -- 分区更快些**

#### 索引的概念？

索引相当于目录，是对某些特定列中的数据进行排序，生成索引表，该列作为WHERE条件时，扫描索引列，根据ROWID快速定位具体记录，提高查询效率。

#### 索引有哪些不同的类型？

主键 唯一 普通 组合 函数 位图 分区索引分为本地和全局

#### 不同的索引一般用在什么场景下？

唯一索引：当某列任意两行数据不相同，建立PRIMARY KEY主键和UNIQUE CONSTRAINT唯一约束时（索引自动建立）。

组合索引：当多列经常一起出现在WHERE条件中，创建索引。

位图索引：当一列有大量重复数据时建立。

函数索引：在WHERE 条件语句中包含函数或表达式时建立。

#### 普通索引和位图索引有什么区别？

普通索引：列的内容没有什么特点，但是经常需要被进行查询创建：

位图索引：列中有非常多的重复的值时候创建。例如某列保存了 “性别”信息创建：

#### 索引在什么情况下会失效？

针对索引使用函数；

索引列算数运算：

索引列隐式转换；

！= 和NULL判断；

LIKE “%\_”百分号在前；

#### 索引的优缺点？

索引优点：创建唯一索引，可以保证数据库中每一行数据的唯一性；可以加快数据的检索速度；可以加速表和表之间的连接；

索引缺点：创建和维护索引要耗费大量时间；索引会占物理存储空间；对表中数据进行操作，索引也要动态维护；

#### 哪些列该建立索引？哪些列不应建立索引？

建立索引：经常需要搜索、排序的列；关联字段；

不该建立索引：很少使用的列；只有很少数据值的列；定义为TEXT,IMAGE,BIT数据类型的列；

#### 唯一索引和主键索引有什么区别

主键是不能自己独立创建，和主键约束一起创建的；

唯一可以和约束一起创建，也可以单独创建。

主键是保证字段不能空不重复，唯一只能保证不重复。

#### 索引中本地索引和全局索引的区别是什么？

本地索引不能创建唯一索引的类型，因为每个小的索引只能保证每个分区的数据是唯一的，不能保证整个表格的数据是唯一的。

全局索引不能创建位图索引的类型，统一管理的索引，无法对整个表格中，每个独立的分区的数据进行识别和判断。

#### 怎么查看一个表格有哪些不同的索引？

select \* from user\_indexes where table\_name='表名';

#### oracle里面，如果有一个sql语句运行很慢，怎么办？\*\*\*\*\*

1. 先查看执行计划，查看句子消耗的资源和内部运行的逻辑，查看语句是否运行复杂等，

查看每个子句运行的过程

2. 给数据量非常大的表格建分区（emp表为例，2个G是8000多万）

3. 给查询的字段进行索引的创建

4. 查看有没有导致索引失效的语句

5. Or会导致表格重复被读取，从而效率低下

6. sql语句频繁的对硬盘进行了读写，可以把语句写成代码块（例如存储过程）让硬盘的读写次数变少

7. sql语句可能会在客户端和服务器之间来回的进行数据的交互，这个时候也可以写成代码块减少客户端和服务器之间的交互次数

8. 还可以使用优化器里面的parallel进行sql语句的并行处理

#### oracle有哪些常见的数据类型？

integer number char varchar2 date blob clob

#### char 和 varchar 的区别？

#### minus的含义是什么？

查询第一个表格有但是第二个表格没有的数据

#### 在左连接里面，如果使用and筛选和用where筛选，区别是什么？

a b

1 1

2 2

3 4

select \* from a left join b on a.id=b.id and a.id=2;

3 2 1

2 2

1 null

3 null

select \* from a left join b on a.id=b.id where a.id=2;

2 2

#### 左边表有身份证信息，右边也是身份证信息，查询左边有但是右边没有的身份证：

select \* from a

minus

select \* from b;

select \* from a where id not in (select \* from b);

select \* from a left join b on a.id=b.id where b.id is null;

#### 经常用到的函数有哪些？\*\*\*

**聚合函数** max min avg sum count

**单行函数**

**数字** abs round trunc floor ceil power mod

**字符串** substr concat instr replace length wm\_concat translate ltrim rtrim lpad rpad

upper lower initcap

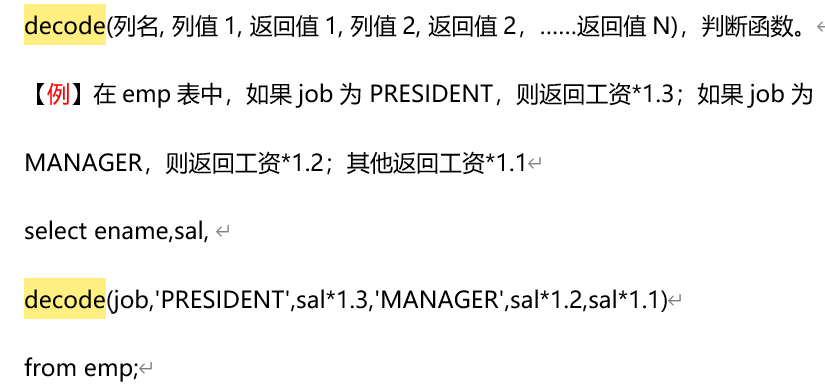
**分析函数（开窗函数/窗口函数）**

排名 row\_number rank dense\_rank

平移 lag lead

**row\_number rank dense\_rank区别**

**逻辑函数** nvl nvl2 decode case...when... pivot



将字符串数字转换成数值型数字：to\_number()

将数字类型转换成字符串：to\_char()

#### 伪列有哪些有什么用？

rowid 利用唯一性进行表格数据的去重操作；索引也是使用rowid进行行数据查询的

rownum 进行表格数据的分页查询

#### 表空间怎么创建、怎么查看、有什么作用？怎么扩展？

create tablespace 名字 datafile '' size xxxM;

select \* from user\_tablespaces;

将表格或者分区和电脑上的数据文件进行绑定的一个数据库对象。

#### 执行计划里面经常要查看哪些内容？\*\*\*\*\*\*\*\*\*

执行计划是一条查询语句在Oracle中的执行过程或访问路径。

执行完一条select语句后，在PLSQL执行计划窗口查看，或按F5即可查看刚执行的这条查询语句的执行计划。

或者在命令窗口输入：

SQL> explain for select \* from emp

SQL> ;

Explained

SQL> select \* from table(dbms\_xplan.display());

主要看以下内容：

1. 执行顺序：缩进最多的先执行；同一级的动作遵循最上最右先执行；
2. 表扫描方式：全表扫描、索引扫描
3. 关联机制：嵌套循环、哈希连接、排序-合并连接
4. 耗费（COST）、CPU耗费：Oracle估计的该步骤的执行成本，用于说明SQL执行的代价，理论上越小越好

#### 什么是视图？优点缺点是什么？

视图就是一张或多张表上的**预定义查询**。

作用：减少子查询的复杂性；提高运行效率；可以仅提供视图数据，提高数据安全性。视图以定义的方式存储在数据库中，不占用表空间。

#### 什么是物化视图？和视图的区别是什么？

视图固定为表称为物化，物化视图是当时建立视图时固定下来的数据，如果原表数据有更新，物化视图中的数据是不变的。

物化视图是一种特殊的物理表，“物化”(Materialized)视图是相对普通视图而言的。

普通视图是虚拟表，应用的局限性大，任何对视图的查询，Oracle都实际上转换为视图SQL语句的查询。这样对整体查询性能的提高，并没有实质上的好处。

#### 存储过程由哪些代码构成？

create or replace procedure 名字(参数 类型)

as

声明

begin

执行

exception

异常

end;

#### 什么时候你会写存储过程？（写存储过实现了哪些功能）

使用存储过程来对sql语句进行优化，减少硬盘的读写和与服务器之间的交互次数；

复杂的逻辑使用存储过程；

一些数据库的固定操作；

表格的数据抽取的ETL操作。

#### 存储过程和函数（存储函数/自定义函数）的区别？

1.返回值的区别,函数有1个返回值,而存储过程是通过参数返回的,可以有多个或者没有

2.调用的区别,函数可以在查询语句中直接调用,而存储过程必须单独调用.

函数一般情况下是用来计算并返回一个计算结果而存储过程一般是用来完成特定的数据操作（比如修改、插入数据库表或执行某些DDL语句等等）

#### 数据仓库是用什么方法来保存历史数据的

数仓仓库保存历史数据通常是在dw层或者是hdw层 创建拉链表（缓慢变化维）来保存历史数据

每次更新数据都会通过update进行逻辑删除，然后将最新的数据插入到拉链表中，并将历史数据更新成无效状态，将最新的数据更新为有效状态

#### 如果oracle运行提示no data found是什么原因造成的？

select into的时候，where筛选的过程没有找到数据

#### 拉链表是什么结构？什么是缓慢变化维？

**拉链表**：是针对数据仓库设计中表存储数据的方式而定义的，顾名思义，所谓拉链，就是记录历史。记录一个事物从开始，一直到当前状态的所有变化的信息。

维度建模的数据仓库中，有一个概念叫Slowly Changing Dimensions，中文一般翻译成“缓慢变化维”，经常被简写为SCD。缓慢变化维的提出是因为在现实世界中，维度的属性并不是静态的，它会随着时间的流失发生缓慢的变化。这种随时间发生变化的维度我们一般称之为缓慢变化维，并且把处理维度表的历史变化信息的问题称为处理缓慢变化维的问题，有时也简称为处理SCD的问题。

**处理缓慢变化维的方法通常分为三种方式：**

第一种方式是直接覆盖原值。这样处理，最容易实现，但是没有保留历史数据，无法分析历史变化信息。第一种方式通常简称为“TYPE 1”。

第二种方式是添加维度行。这样处理，需要代理键的支持。实现方式是当有维度属性发生变化时，生成一条新的维度记录，主键是新分配的代理键，通过自然键可以和原维度记录保持关联。第二种方式通常简称为“TYPE 2”。

第三种方式是添加属性列。这种处理的实现方式是对于需要分析历史信息的属性添加一列，来记录该属性变化前的值，而本属性字段使用TYPE 1来直接覆盖。这种方式的优点是可以同时分析当前及前一次变化的属性值，缺点是只保留了最后一次变化信息。第三种方式通常简称为“TYPE 3”。

#### **Oracle数据库的端口号是多少**

默认是1521

#### Oracle 数据库日常维护？

检查Oracle实例状态，检查Oracle服务进程，检查Oracle监听进程；

#### 什么是伪列，有哪几种，区别在那里？

1. ROWID:表中的每一行在数据文件中都有一个唯一物理地址，ROWID伪列返回的就是该行的物理地址,使用ROWID可以快速的定位表中的某一行,所以可以用来查重。
2. ROWNUM，为结果集中每一行标识一个行号，第一行返回1，第二行返回2，以此类推，通过ROWNUM伪列可以限制查询返回的行数。

#### 如何使用ROWID去重？

DELETE FROM 表名 别名WHERE ROWID NOT IN(SELECT MIN(ROWID) FROM 表名 别名 GROUP BY 列名)

如果想通过ROWID去重，那么在 delete 重复数据时，需要group by 表的所有字段。如果只group by 表的个别字段，那么会造成误删除。

#### 如何分页查询？

可以用伪列rownum进行分页查询

--举例：第一页显示前5行

select ROWNUM,e.\* from emp e

where rownum<=5--查询前5行，不能大于！

---第二页显示6-10

select \* from

(

select ROWNUM rn,e.\* from emp e

) t where t.rn>5 and t.rn<11

--第三页显示11-14

select \* from

(

select ROWNUM rn,e.\* from emp e

) t where t.rn>10 and t.rn<15

#### SQL执行顺序？

（1）from来自不同数据源的数据；

（2）where基于指定的条件对记录行进行筛选；

（3）group by子句将数据划分为多个分组；

（4）使用聚合函数进行计算；

（5）使用having子句筛选分组；

（6）计算所有的表达式；

（7）select 的字段；

（8）使用order by对结果集进行排序。

#### 如何对重复数据去重？

distinct、exists、group by 、rowid，union。其中distinct性能最差

#### 常用的SQL函数有哪些？

字符串函数、日期函数、数值函数、NVL、decode、case when等。

#### 常见的分析函数（开窗函数）有哪些？

聚合类分析函数：MAX(),MIN(),SUM(),AVG(),COUNT()

排名类的分析函数：RANK()1224,DENSE\_RANK()1223与ROW\_NUMBER()1234：--分析函数中排序

位移分析函数：LAG()与LEAD()：求之前或之后的第N行

#### 如何列转行、行转列？

**列转行：**用分析函数lead/lag；用DECODE；用CASE WHEN;部分关联；

**行转列：**pivot(聚合函数(有数据的列) for 想转置的列名 in (该列名中的数据));

#### 分析函数和聚合函数的区别？

◎普通的聚合函数用group by分组，每个分组返回一个统计值，

◎而分析函数采用partition by分组，并且每行都可以返回一个统计值，

◎写法上也有区别，分析函数要带 over（）。

#### 分析函数的使用需要注意什么？

**聚合类分析函数：**

---over 之后的括号里可以什么都不写

---如果写了partition by意味着分组统计

---如果加了order by 是累计统计

**排名类的分析函数：**

----over 之后括号里必须要有order by，

----可以没有partition by，可以不分组

**LAG()与LEAD()位移类分析函数：**

----over 之后括号里必须要有order by，

----可以没有partition by，可以不分组

#### in与not in,exists与not exists的区别以及性能分析？\*\*\*

EXISTS(查询结果集)：查询结果集有记录则成立，否则不成立，可以加不等式；

IN操作符来查询其列值在指定的列表中的行，只能查询值；

性能主要还是看exists和in后面的表的数据量，数据量一致时，exists性能好。

#### ON和WHERE中写过滤条件的区别是什么？

不管是WHERE 还是ON，Oracle都会把能**过滤的条件**先过滤掉，再关联。但两者区别在于，如果是内关联，两种结果相同，如果是外关联，结果会不同，ON会保留主表的所有信息，而WHERE可能会过滤掉部分主表信息。

select \* from dept d left join emp e on d.deptno=e.deptno and d.deptno=40;

select \* from dept d left join emp e on d.deptno=e.deptno where d.deptno=40

#### **count(\*) 和 count(1)和count(列名)区别？**

**从执行结果来说：**

count（1）和count（\*）之间没有区别，因为count（\*）count（1）都不会去过滤空值，但count（列名）就有区别了，因为count（列名）会去过滤空值。

**从执行效率来说**：

（1）如果列为主键，count(列名)效率优于count(1) ；如果列不为主键，count(1)效率优于count(列名) ；

（2）如果表中存在主键，count(主键列名)效率最优；

（3）如果表中只有一列，则count(\*)效率最优；如果表有多列，且不存在主键，则count(1)效率优于count(\*)；

#### 什么是同比和环比？如何用SQL实现同比环比的计算？

环比=（现阶段 - 同一周期上一阶段）/同一周期上一阶段

同比=（现阶段 - 上一周期相同阶段）/上一周期相同阶段

可以使用LEAD和LAG函数实现

#### Union和union all 的区别？

1)UNION和UNION ALL都是将两个结果集合并为一个，但是UNION在进行表链接后会筛选掉重复的记录，Union All不会去除重复记录。

2)Union将会按照字段的顺序进行排序；UNION ALL只是简单的将两个结果合并后就返回。

3)从效率上说，UNION ALL 要比UNION快很多；

#### Oracle where 子句中多个and和or时，执行顺序是怎么样的？

三个逻辑运算符优先级：NOT>AND>OR

#### 知道varchar与varchar2的区别吗？空值怎么转换及区别？

1)varchar2所有字符都占两字节处理(一般情况下)，varchar只对汉字和全角等字符占两字节，数字，英文字符等都是一个字节；

2)VARCHAR2把空串等同于null处理，而varchar仍按照空串处理；

#### Delete和truncate的区别，和drop又有什么区别？

1)在速度上，一般来说，drop> truncate > delete。

2)delete 是DML 操作，是一行行的删除数据，可以回滚,可以恢复，可以筛选数据, 如果想删除部分数据用delete，注意带上where子句；

3)truncate 是DDL操作，是删除整个表格然后重新创建一个新的，不能回滚不能筛选。

#### 知道怎么授权吗？取消授权呢？

常用的系统权限集合有以下三个:CONNECT(基本的连接), RESOURCE(程序开发), DBA(数据库管理)

**1)GRANT 授予权限：**GRANT CONNECT/RESOURCE TO 用户名;

**2)REVOKE 回收权限:** REVOKE CONNECT/RESOURCE FROM 用户名;

#### 什么是数据字典？如何查询系统数据字典？

查看对应字段的具体含义，字段类型，长度，约束，注释等。

select a.tablespace\_name,--表空间名称

a.table\_name,--表名

b.column\_name,--字段名

b.data\_type,--字段类型

b.data\_length,--长度

c.comments--注释

from

user\_tables a,

user\_tab\_cols b,

user\_col\_comments c

where a.table\_name = b.table\_name

and b.column\_name = c.column\_name

and b.table\_name=c.table\_name

and a.table\_name='EMP'

#### Oracle SQL语句由什么命令组成？

**数据定义语言**（DDL）：包括CREATE（创建）ALTER（修改）DROP（删除）等。

**数据操纵语言**（DML）：包括INSERT（插入）UPDATE（更新）DELETE（删除）等。

**数据查询语言**（DQL）：包括基本查询语句SELECT和Order By子句、Group By子句等。

**事务控制语言**（TCL）：包括COMMIT（提交）命令) ROLLBACK（回滚）命令。

**数据控制语言**（DCL）：GRANT（授权）命令) REVOKE（撤销）命令。

**二、表空间和数据库对象**

#### 什么是事务？事务有哪些特点？

事务就是多条sql语句组成的工作逻辑单元

事务的特点：原子性、一致性、隔离性、持久性

事务对SQL 语句进行分组，使它们要么全部提交 ( 这意味着它们被应用到数据库 ) ，要么全部回滚 ( 这意味着它们从数据库撤消 ) 。

只要有一条 SQL 语句执行失败，则已执行的 SQL 语句会回滚到执行之前的状态，这样就保证了数据库数据的一致性，不至于产生混乱的数据。

•原子性  Atomicity

事务必须是原子工作单元，对其进行的数据修改，要么全都执行，要么全都不执行。

强调事务的不可分割 。

例如： 银行转账， A 向 B 转账 1000 元， 要在A 账户上 减少 1000 元，同时要在 B 账户上 增加 1000 元。 并且记录一条转账记录。 要么同时执行，要么都不执行更改，以确保整个事务是一个原子工作单元。

•一致性  Consistency

事务将数据库从一个一致状态转移到另一个一致状态。

强调事务执行前后, 数据库的完整性保持一致 。

例如， 银行转账，A 向 B 账号转账 1000 元，是不允许看到 A 的账号减少 1000 ， B 的账号还没来得及增加 1000 的中间状态。

•隔离 性 Isolation

一个事务不会看到另外一个还未完成的事务产生的结果。每个事务就像在单独、隔离的环境下运行一样。

强调事务并发访问 ， 一个事务的执行 ， 不应该受到另一个事务的打扰 。

例如，一个用户正在更新员工表。其他用户不会看到对员工表所做的未提交更改。因此，在用户看来，事务似乎是串行执行的。

• 持久 性  Durability

提交的事务所做的更改是永久性的。 不会因为系统失败而丢失。强调事务结束 ， 数据永久保存数据库中 。事务完成后，数据库通过其恢复机制确保事务中的更改不会丢失。

#### 什么是锁，什么是死锁？

锁就是某个用户在更新、修改数据时，锁定当前表或记录，不允许其他用户修改。

死锁就是两个会话，每个会话持有对方想要的资源，因争夺资源而造成的互相等待的现象。

#### 如果发生死锁，如何处理：

首先要有管理员权限，然后在plsqldev工具中打开：工具-会话窗口

找到带锁的会话，点击关掉（kill）这个会话。

#### 什么是表空间？系统常见的表空间有哪些？

表空间就是一个或多个数据文件的集合，ORACLE所有的数据对象都存放在指定的表空间中，但主要存放的是表，所以称作表空间。系统常见的表空间有系统表空间、TMEP表空间、用户表自定义空间。

系统表空间：存放系统数据，例如：数据字典、视图；

TMEP表空间：存放临时缓存数据；

用户表定义空间：存放用户创建的表数据；

#### 什么是同义词？有什么作用？

同义词是数据库对象的一个别名。

作用：方便其他用户访问数据库对象。

#### 什么是序列？常用在什么地方？

序列是用来生成连续的整数数据的对象，可以指定生成逻辑（开始整数、结束整数、是否循环）。

序列常用来作为主键中增长列。

#### 表分区的概念？

表分区就是将表中数据在物理上存放在多个表空间，这样查询数据时不需要扫描整张表的数据文件

#### 什么时候使用表分区？

表的数据量特别大；表中包含历史数据，新的数据被增加到新的分区中。

#### 表分区的优点和缺点？

优点：改善查询性能；增强数据可用性、可靠性；维护方便；

缺点：已经存在的表无法直接转化为分区,需要维护。

#### 如何把一张没有分区的表转化为分区表？

**使用create table as select方式创建一个分区表，然后将普通表重命名，然后对新的分区表创建所需索引，重命名等。**

CREATE TABLE T\_NEW

()

...--带分区

AS SELECT \* FROM T;

RENAME T TO T\_OLD;

RENAME T\_NEW TO T;

1. **PL/SQL程序设计**

#### 什么是游标，如何使用？

游标是指向查询结果集的一个指针，通过游标可以将查询结果集中的记录逐一取出，并在PL/SQL程序块中进行处理。

具体采用OPEN、FETCH和CLOSE语句来控制游标。或者使用FOR x in loop的方式。

#### 游标的属性有哪些？

游标属性：%FOUND和%NOTFOUND

%FOUND:

用于判断游标是否从结果集中提取数据。如果提取到数据，则返回值为TRUE，否则返回值为FALSE。

%NOTFOUND:

该属性与%FOUND相反，如果提取到数据则返回值为FALSE；如果没有，则返回值为TRUN。

#### 动态sql如何实现变量绑定？好处是什么？

如果动态语句是SELECT语句，可以把查询的结果保存到INTO后面的变量中。如果动态语句中存在参数，USING为语句中的参数传值。动态SQL中的参数格式是：[:参数名]，参数在运行时需要使用USING传值。

sqlstr:='select ename from emp where empno=:id';---动态sql内的变量（：加变量名字）

EXECUTE IMMEDIATE sqlstr into sename using sempno;

好处是可以实现软解析

#### 如何创建存储过程？

包括过程的名称、过程使用的参数以及过程执行的操作

#### 存储过程参数的区别？

IN表示传入参数，不可以被赋值，OUT表示传出参数，可以被赋值，IN OUT表示传入传出参数，可以传入值，可以被赋值，可以返回值

#### 常用的异常都有哪些？

|  |  |  |
| --- | --- | --- |
| **异常名称** | **异常码** | **描述** |
| DUP\_VAL\_ON\_INDEX | ORA-00001 | 试图向唯一索引列插入重复值 |
| INVALID\_CURSOR | ORA-01001 | 试图进行非法游标操作。 |
| INVALID\_NUMBER | ORA-01722 | 试图将字符串转换为数字 |
| NO\_DATA\_FOUND | ORA-01403 | SELECT INTO语句中没有返回任何记录。 |
| TOO\_MANY\_ROWS | ORA-01422 | SELECT INTO语句中返回多于1条记录。 |
| ZERO\_DIVIDE | ORA-01476 | 试图用0作为除数。 |
| CURSOR\_ALREADY\_OPEN | ORA-06511 | 试图打开一个已经打开的游标 |

#### 如何找到存储过程哪个环节执行的快慢？

用日志是用来追溯问题的，记录整个程序的运行情况，知道哪个环节报错了，记录每一步花了多少时间，判断哪一步性能不好，从而对程序进行修改和优化。

#### 包的组成？

一个程序包由两部分组成：包定义和包体。其中包定义部分声明包内数据类型、变量、常量、游标、子程序和函数等元素，这些元素为包的共有元素。包主体则定义了包定义部分的具体实现。

#### 什么是全局变量？

在包定义部分声明的变量在包体里的存储过程或者函数都可使用，这个是全局的，在存储过程或者函数内部定义的只能这个存储过程或者函数使用，是局部的。

**四、Oracle优化**

**重点：SQL语句优化、索引优化、表分区优化、并行优化！**

**同时关注执行计划！**

#### 常用的SQL语句调优？

1.避免索引失效；

2.多使用COMMIT提交数据；

**3.in,not in 用exist,not exist 替换，少用JOIN；**

**4.临时表；使用WITH..AS语句，对某段SQL建立临时表；拆分步骤；**

**5.DISTINCT 性能最差的去重，尽量用group by替代**

**6..使用SQL内置函数加快查询速度，例如使用decode函数替换case when；**

7.避免在select 后使用'\*' 符号；

8.连接和分组时先使用where条件过滤

**9.避免笛卡儿积；**

10.删除全表数据时TRUNCATE替换DELETE;

#### 优化器是什么？有哪几种？

优化器是SQL分析和执行的优化工具，负责生成、制定SQL的执行计划。

种类：RBO基于规则的优化器；CBO基于代价的优化器（cost、time、cpu）；

#### 表的扫描方式有哪些？优缺点是什么？

主要有全表扫描、索引扫描

优缺点：全表扫描可直接读取表中所有行，适合数据量小或者数据量大返回结果多；

索引扫描，查询速度快，耗费资源少，先扫描索引得到对应的ROWID，然后通过ROWID定位到具体的行读取数据；适合数据量大，返回值少的表；

#### 表的关联方式有哪些？优缺点是什么？

嵌套循环、哈希连接、排序-合并连接

优缺点：

嵌套循环分为驱动表和匹配表，从驱动表获取一行数据，在匹配表找符合条件的行；确定合适驱动表，建立索引，执行效率会很高；

哈希连接，先生成哈希函数，对两张表生成哈希值，利用哈希值建立连接。在数据量大，无索引时，比嵌套效率高，适合等值连接；

排序-合并，先排序，再连接；针对数据量大，无索引，需要排序时运用，适用不等式连接；

#### 什么是并行，如何通过并行增加数据处理速度？加多少并发？

并行就是采用parallel技术，把一个大的任务分为若干小任务，同时启用n个进程/线程，同时处理这些小任务。通过消耗CPU资源增加数据处理速度。

**适用parallel的两个条件：**

1）**大的任务**，如全表扫描大表，小任务自己完成比派发省事

2）**系统有足够的资源(cpu/io)**

换句话说，并发是在系统资源充足，用户少的系统上，为了充分利用系统资源以提高任务处理速度而设计的一种技术。

#### 如何看hints 是否被调用？Hints原理

查看执行计划有个**PX**字段代表;

在绝大多数情况下执行计划会选择正确的优化器，减轻了DBA的负担。但有时它也聪明反被聪明误，选择了很差的执行计划，使某个语句的执行变得奇慢无比。此时就需要DBA进行人为的干预，告诉优化器使用我们指定的存取路径或连接类型生成执行计划，从而使语句高效的运行；

#### 常见的hints优化器有哪些？

/\*+ PARALLEL(表名,并行数) \*/ --指定开启多少个并行

/\*+ INDEX(表名 索引名) \*/ --指定索引

/\*+ FULL(表名) \*/ --指定全表扫描

/\*+ USE\_NL(表名1 表名2) \*/ --指定用NESTED LOOP连接

/\*+ USE\_HASH(表名1 表名2) \*/ --指定用HASH连接

/\*+ USE\_MERGE(表名1 表名2) \*/ --指定用SORT MERGE JOIN(不等式运用）/\*+ LEADING(表名1 表名2) \*/ --指定表1作为驱动表

/\*+ APPEND \*/ --数据直接插入到高水位上面(与insert连用)

#### 有两张很大的表关联，怎么做才能关联性能好？

1.首先要建立适当的索引。SQL在索引字段不要加函数，保证索引起效。如果是复合索引注意在SQL的顺序。如果已经存在索引，建议你先重建索引，因为大数据表的索引维护到了一个阶段就是乱的，一般建议重建。建立好的一般可以获得几十倍的速度提升。

2.最大数据量的表放在最前，最小的表放在最后面。SQL是从最后面开始反向解析的。

3.其次是要把最有效缩小范围的条件放到SQL末尾去。尤其是主键或者索引字段的条件。

4.为两个表设计合理的表分区，然后分别对应关联两个表的分区数据，再用union all把各个连接结果叠加起来；

#### 大表和小表连接如何进行优化？

将小表作为驱动表，大表作为匹配表，关联字段建索引。

#### Oracle内部执行机制?

**1）语法分析**，分析语句的语法是否符合规范，衡量语句中各表达式的意义。

**2）语义分析**，检查语句中涉及的所有数据库对象是否存在，且用户有相应的权限。

**3）视图转换**，将涉及视图的查询语句转换为相应的对基表查询语句。

**4）表达式转换**， 将复杂的 SQL 表达式转换为较简单的等效连接表达式。

**5）选择优化器**，不同的优化器一般产生不同的“执行计划”；

**6）选择连接方式，**ORACLE 有三种连接方式，对多表连接 ORACLE 可选择适当的连接方式。

**7）选择连接顺序**,对多表连接 ORACLE 选择哪一对表先连接，选择这两表中哪个表做为源数据表。

**8）选择数据的搜索路径**，根据以上条件选择合适的数据搜索路径，如是选用全表搜索还是利用索引或是其他的方式。

**9）运行“执行计划”。**

#### 执行计划执行顺序

先执行缩进量最多步骤，如果缩进量一直，优先执行上面的。

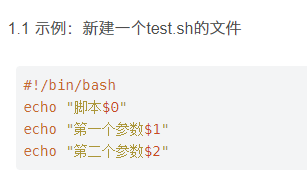
#### delete删除数据，commit提交后，如何找回

delete删除数据之后，默认会将数据放在oracle对应的表格的回收站中。（oracle的闪回机制 flush）

# linux和shell

#### shell怎么传参

接收参数：采用$0,$1,$2..等方式获取脚本命令行传入的参数，值得注意的是，$0获取到的是脚本路径以及脚本名



执行shell脚本传入参数：



#### 批量杀死名字带abc的进程

kill -9 `ps -aux | grep abc | awk '{print $2}]'`

#### **Linux常用的一些命令**

文件文件夹有关：mkdir touch cp mv rm head tail more cat grep awk sed cut sort wc uniq

软硬件相关：ps netstat kill top free df service chkconfig

权限：useradd userdel chmod su

定时任务：crontab

#### 怎么定时

crontab -e

crontab -l

#### Linux怎么查看具体的进程信息

ps -aux

#### 怎么替换一个txt文本中的，为；？

sed -i '1,$s/,/;/g' xx.txt

#### 怎么创建一个新的sh脚本，怎么使用

创建sh脚本使用touch或者是vim命令。

使用 sh bash 或者是路径加上文件名来使用。

#### 如何把10万行数据按照每1000行切割成100个小文件。

后缀为字母：split -l 1000 -a 2 1.txt a\_

或

后缀为数字：split -l 1000 -a 3 -d 1.txt a\_

# 大数据部分

#### hive数据仓库和其他的数据库有什么区别？

1. hive数据库是使用hdfs存储数据的，oracle数据库使用.dbf文件存储数据

2. hive数据库使用mr引擎计算数据，oracle有自己自带的executor进行计算

3. hive的语句运行延迟比较高，oracle延迟很低

4. hive支持的容量比较大，可以支持TB PB，但是oracle只能支持GB

5. hive数据库没有约束条件的部分，oracle有约束

6. hive没有索引，oracle有索引

#### 传统数仓和大数据数仓的区别---OLTP和OLAP的区别

* 传统数仓主要是结构化的数据，大数据数仓有结构化，半结构化，非结构化的数据
* 传统数仓数据量为GB，TB；大数据数仓数据量为pb级
* 传统数仓用oracle存储，大数据数仓用hdfs，hbase存储
* 传统数仓主要处理离线数据，大数据数仓既可以处理离线数据，也能处理实时数据

#### 你们使用的hadoop是什么环境什么版本的？

hadoop 开源版 3.1.1

Hive 开源版3.1.2

MySQL 5.7

Sqoop：1.4.7

hadoop cdh版本 5

#### 分桶的作用是什么？

进行表格连接的查询加速；加快表格的抽样查询。

#### 分桶的数量你是如何决定的？

总的文件大小/(block size\*2) = 分桶数量

#### hive里面经常用到的函数有哪些？

**聚合 sum avg min max count**

**单行**

**数字 abs round floor ceil power**

**字符串 substr concat concat\_ws replace instr split translate upper lower initcap lpad rpad length trim ltrim rtrim get\_json\_object**

**时间日期 current\_timestamp unix\_timestamp from\_unixtime add\_months months\_between date\_add datediff last\_day**

**数据类型转换 cast**

**数组 array size collect\_list collect\_set explode**

**映射 map map\_keys map\_values**

**分析**

**排名 row\_number rank dense\_rank**

**平移 lag lead**

**逻辑 nvl case...when...**

#### 进行数据的同比和环比有什么方法？

lag lead

表连接 select \* from a join b on a.year=b.year+1 and a.month=b.month;

#### hive表连接和其他数据库有什么区别？

hive使用left semi join替换子查询嵌套；表连接不能使用on后面的非等值查询。

#### hive里面经常用到的几个不同的排序分别是什么，有什么区别？

order by 是整个表格的数据当成一个进程进行整体的排序

sort by 可以根据 mapred.reduce.tasks进行分组的设置，每个组分别进行排序

cluster by 只能对该列进行升序排序

distribute by 本身没有排序功能，需要和sort by一起

#### 表格的存储格式有哪些？

textfile

sequencefile

rcfile

orc

**Parquet**

#### 项目中经常用的结构有哪些？

如果表格比较小会使用textfile，中大型的表格使用orc

#### 表格的压缩格式你们用的什么？

textfile gzip

orc zlib

sequencefile BLOCK RECORD

#### hive的优化

1. 可以通过explain查看一下sql语句运行的步骤、逻辑、使用方法、数据的类型、前后的顺序等等内容

2. 减少查询过程中使用到的数据量

分区 分桶 不使用select \*

先筛选数据再计算或者再连接表格

3. 调整表格的存储格式

4. 根据表格的结构设置map或者reduce的数量

#### 多大的表算小表

2G以下的表算小表

**2百万行**

**你操作过最大的表有多大**

**上亿行**

#### 怎么设置map？

map的数量和两个开关有关，有一个merge的合并开关和一个map.tasks的数量开关，

使用 表格的总数量大小/tasks和merge设置取最小值，计算每一个map需要运行的数量大小，查看map运行的数量是否有很大的明显的差距

#### 怎么设置reduce？

如果我们的表格是列比较少但是行比较多的大表，那么通过reduce.tasks设置表格分桶的数量，

然后通过 select \* from t distribute by 列; 进行表格的分桶。

#### 什么叫做数据倾斜呢？

在进行表格的查询和计算过程中，如果有的进程计算的数据很大，有的进程计算的数据比较少，有的部分进程就需要等待其他数据大的进程运行完成才能结束。所以计算过程中资源的分布发生了不合理的现象。

#### 有没有遇到过数据倾斜的问题？数据倾斜的表现是什么？

reduce的进程在日志里面卡在了99或者100的部分不动了。

#### 哪些情有哪些情况会导致数据倾斜呢？

1. map和reduce的设置不合理

2. 对表格进行了去重的结果查询

3. 大小表格进行数据的查询

4. key值分配不均

5. 进行数据查询的时候有大量的空值数据参与了数据的运算

#### 怎么解决数据倾斜的现象呢？

1. 设置 mapred.map.tasks; mapred.reduce.tasks;

2.在句子中尽可能少的使用去重统计

3.

-- 3.1使用hive里面的优化器 /\*+ mapjoin(小表的名字) \*/

-- 3.2调整表格的前后顺序，hive里面，永远读取Join前面的表格

-- 3.3对表格设置分桶

4. 使用hive里面的万能开关，来解决异常数据的问题

set hive.groupby.skewindata=true; --万能方法

set hive.map.aggr=true; --任务执行中合并文件的大小

set hive.merge.size.per.task=256000000;--调整每个map处理数据量的次数，

set mapred.map.tasks=10; ----这里设置的是每一个map要处理的map数量

5. 填充空值，用字符串+随机值进行空值的填充

nvl(被计算的列, concat('rand\_',rand()))

#### mapjoin优化器的原理是什么？

将小表格的数据放入到内存里面，让大表去读取内存中小表的数据进行数据的匹配和连接

**mapjoin**是将小表格先读取到内存中，然后用大表格去匹配内存中小表的数据。

#### 两表连接后能得出什么结果

**A id B id**

**1 1**

**2 2**

**3 4**

**select \* from A left join B on a.id=b.id and a.id=2;**

**2 2**

**1 null**

**3 null**

**select \* from A left join B on a.id=b.id where a.id=2;**

**2 2**

#### a表有4条数据 b表有3条数据 请问左连接有几条数据

**最少是4条 最多是6条**

**1 5**

**2 5**

**3 5**

**4**

**1 1**

**2 1**

**3 1**

**4**

#### hadoop有哪三大组件？

hdfs ： hadoop的分布式文件管理系统

mapreduce ： 数据的计算引擎

yarn ： 资源管理和调度系统

#### hadoop 平台，你用过和知道哪些不同的组件？

离线的部分：sqoop yarn hdfs mapreduce hive

实时的部分：flume(日志信息的收集) kafka(消息队列的处理) hbase(一种列式存储的数据库) spark(基于内存的计算引擎) flink(流式处理的计算引擎)

#### hadoop里面，hdfs数据块是多大一块？

128M

#### 数据默认保存几份？

3份

#### hdfs里面由哪几个组件构成？

datanode namenode secondarynamenode

#### hdfs里面的几个组件，分别有哪些功能和作用？

secondarynamenode：服务器数据的收集，将信息传递给namenode

namenode：负责和客户端进行沟通

datanode：负责存储数据

#### hadoop的基础服务有哪几个？

datanode namenode secondarynamenode jps resourcemanager nodemanager

#### hdfs里面，写入数据（上传文件）和读取数据（下载文件），过程流程和原理是什么？

**读取数据：**

1.客户端申请某个位置的文件或者数据

2.namenode响应申请，并且将文件和数据所在的datanode节点信息列表返回给客户端

3.客户端根据节点信息去向datanode申请数据的读取

4.datanode响应成功给客户端

5.客户端开始申请读取block1

6.datanode返回block1的数据

7.持续申请后面的其他block数据

8.datanode持续的返回剩下的其他数据

**写入数据：**

1.客户端要申请写入一个数据

2.namenode审核文件和数据的合法性

3.namenode返回允许的响应

4.客户端开始申请写入

5.namenode返回datanode的节点信息

6.客户端找到datanode开始申请写入数据

7.datanode同意进行数据写入

8.客户端开始上传数据

8.1 datanode开始向其他的datanode申请备份

8.2 其他的datanode同意备份

8.3 开始备份

8.4 备份完成

9. datanode回应客户端表示写入成功

#### hdfs里面，常用的操作命令有哪些？上传、下载、合并等等。

hadoop fs -put

hadoop fs -get

hadoop fs -appendToFile

#### 加载数据到数据库的表，使用什么方法？

load data inpath 'hdfs路径' into table 库名.表名 partition(分区字段=分区值);

load data local inpath 'linux路径' into table 库名.表名 partition(分区字段=分区值);

hadoop fs -put linux文件的位置和名字 表格在hdfs的路径

#### 如果现在，做数据的增量抽取，如何在重复抽取的过程中，避免出现重复的数据？

增量抽取的数据，会放在每天的分区里面中，然后通过删除今天的分区达到避免重复数据出现的效果。

#### 如何删除分区呢？

alter table 表名 drop partition(分区字段=’分区值‘);

load data inpath '路径' into table 库名.表名 partition(分区字段=分区值);

insert overwrite table 表名 partition(分区字段=分区值) select \* from 另一个表名;

#### 如果某个数据有问题，如果更新这个数据？

1. -get 这个文件，修改文件本身，删除hdfs原文件，在-put上传上去

2. 通过select语句修改数据本身写入另一个表，然后删除原表，重新命名新表

3. 如果是orc类型的分桶表格，通过打开ACID的方法，进行数据的update更新

#### 内部表和外部表的区别？

1. 内部表是通过create table创建的，外部表是create external table进行创建的

2. 外部表需要通过location指定表格映射的文件夹

3. drop table的时候，内部表会删除文件夹、结构和数据，外部表只会删除mysql中存储的表格的结构

4. 一般像日志和埋点数据这种数据量非常大的表格，都会保存在外部表中，计算结果会保存在内部表中

#### 动态分区和静态分区的区别？

1. 静态分区可以load data和insert两种方式添加数据，动态只能通过insert的方式添加数据

2. 静态需要自己指定分区的内容，动态是通过查询结果自动指定和分配的

3. 动态需要打开动态分区和非严格模式的开关

**set hive.exec.dynamic.partition=true;**

**set hive.exec.dynamic.partition.mode=nonstrict;**

4. 一般情况下都是使用静态分区，动态的处理数据的速度相对慢一点

#### 为什么分桶可以对表格的联合查询加速呢？

因为分桶可以减少表格笛卡尔积的数量

#### 分区和分桶的区别？

1. 分区是通过partitioned by (新字段) 进行分区设置的，分桶是通过clustered by (已有字段) 进行分桶设置的

2. 分区是以文件夹的方式分开保存数据的，分桶是分成不同的文件保存数据的

3. 分区是在查询分区字段的时候加快查询速度，分桶是对分桶字段进行表格的联合查询的时候进行联合查询加速的

#### 怎么进行数据的抽取？

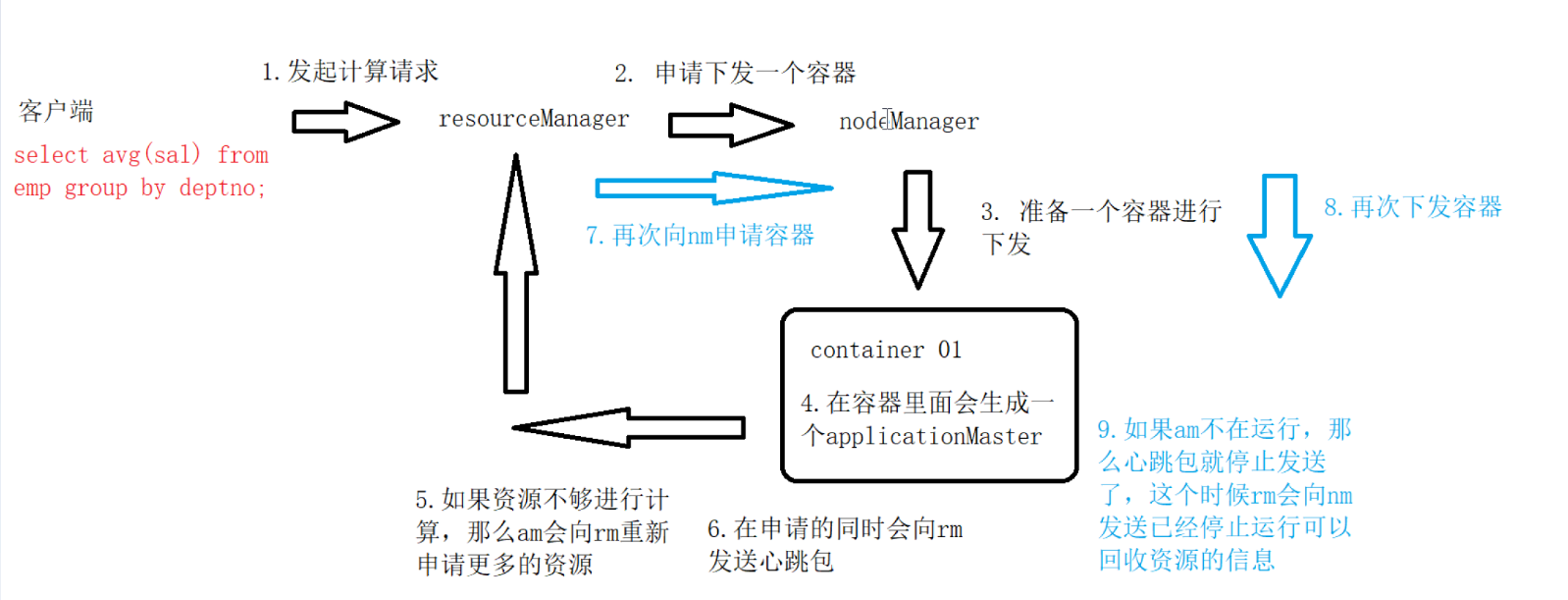
在hive数据库中，可以通过sqoop进行数据的抽取，有全量和增量两种不同的方法，增量一般是通过lastmodified的方式进行数据抽取的，也可以通过分区表的方式抽取数据

#### yarn的原理和流程？

**ResourceManager：**在整个系统中，有且仅有一个的服务，负责调配Container

**NodeManager**：负责启动和管理以及监控Container

**AppMaster：**进行数据计算  **Container：**容器，一部分服务器的资源



#### mapreduce的原理和流程是什么？

input：先获取在计算过程中所需要的数据

split：对大的数据进行切割的操作，将数据分成一块块的

map：数据的映射。将分配好的数据，给到不同的进程去运行

shuffle：每个进程单独的对自己拿到的数据进行计算（对每个数据进行数据拆分(分区)并且进行排序，保存在硬盘上，从硬盘上读取数据，并且进行计算，结果再次排序）

reduce：将单独的数据进行总体汇总的计算过程

finalize：将计算结果输出

#### shuffle的基本流程？

数据进行分区--sort排序--存入硬盘--从硬盘读取数据--聚合计算等等--数据排序

#### 表连接（喜欢问左连接）：

1 1

2 2

4 3

a left join b on a.id=b.id and a.id=2

2 2

1 null

4 null

a left join b on a.id=b.id where a.id=2

2 2

#### 分析函数：

row\_number -- rank -- dense\_rank -- lag -- lead

#### 时间的方法：

获取当前时间：from\_unixtime(unix\_timestamp()) current\_timestamp

#### hive的优化

-- 尽可能减少查询和运算过程中的数据量

分区、分桶、查询列不要使用\*、先筛选再聚合和连接表。

-- 修改表格存储的结构和压缩的方法

小表：textfile gzip

中大表：orc zlib

-- 设置不同的map和reduce数量

-- 有一些不同的hive数据库的set开关

aggr 聚合开关

parallel 并行开关等

#### Hadoop的构架

使用hdfs进行数据的存储

使用mapreduce进行数据的计算

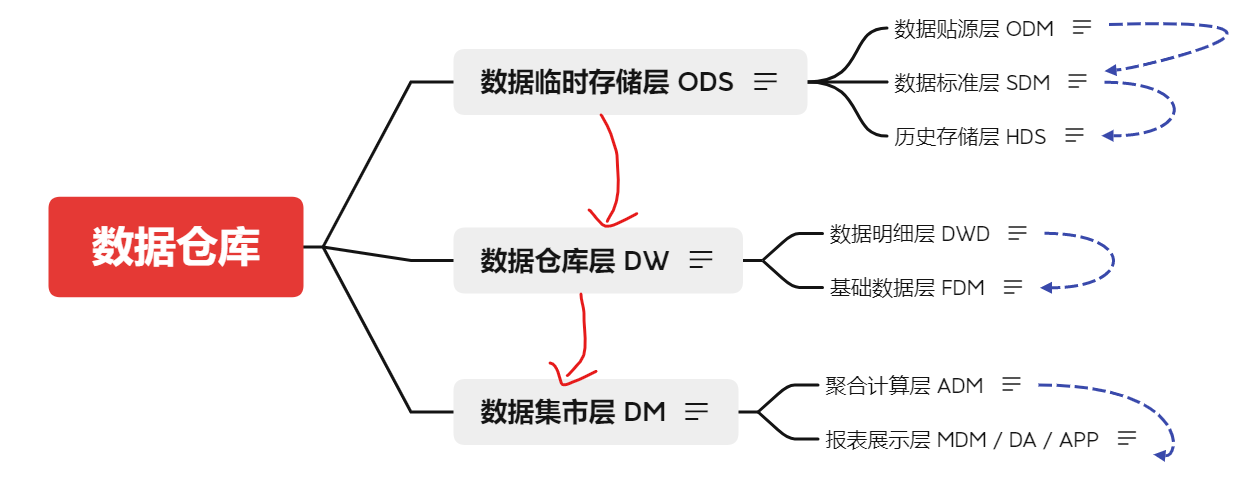
使用yarn进行资源的调度

使用hive编写sql语句

使用mysql存储表格的元数据结构

#### 数据库分层

业务数据的来源：业务系统



#### merge into的基本语法

将目标表格和原表格进行主键的对比，如果主键一致，就更新原表数据到目标表格中，如果主键不一致，就将原表的数据插入到目标表格中。

#### 如果平常数据量增加都很正常，然后现在数据增长幅度突然剧增，要从哪几个角度去思考到底哪里出现了问题？

如果平常数据量增加都很正常，那么我的逻辑控制、变量的设置、数据的获取方式应该都没有什么问题，如果今天突然新增了数据，首先先去确认表格的数据是否都是重复的数据、是否都是空的数据、是否都是异常数据等等，然后往上游的系统进行数据的追溯，查看上游给我们的数据本身是否有问题，如果上游数据很存储数据一致，就找上游协调问题，如果上游数据和存储数据不一致，那么就查找是否有特殊数据导致逻辑出现问题。

还可以查看其他的表格，看是否也有突增的数据，考虑是否是调度的工具出现了问题。

#### Hadoop体系中有哪些数据处理引擎，他们之间的差异有哪些

mapreduce：进程级别的计算引擎、计算过程比较稳定，但是消耗资源比较多，基于硬盘进行计算的

spark： 线程级别的计算引擎，过程没有mr稳定，但是消耗资源少，基于内存计算的

flink：流式计算引擎，可以支持非常大的用户并发计算

#### 一个分区表，一个非分区表，如何用非分区表中的字段作为分区表的分区值？

直接做动态分区就好了，使用 insert overwrite table 分区表 partition(分区字段) select xxxx from 非分区表;

#### linux 怎么创建文件和文件夹？

文件 touch

文件夹 mkdir

#### 怎么找出HDFS中文件的路径？

hdfs里面没有直接查找文件的方法，但是可以通过元数据库例如Mysql里面有一个sds表格，通过Like模糊查询这个文件的名字，然后找到它所在的位置。

#### hive怎么动态的筛选出今天之前的数据?

(current\_timestamp,-1)

dt<from\_unixtime(unix\_timestamp(),'yyyy-MM-dd')

#### hive中 一个表格有ID 和登陆时间，怎么找出昨天登陆次数两次的用户？

提取出年月日的数据，将年月日和用户的Id共同分组，找出组的行数等于2的就可以了

#### hive 怎么截取字符串最后两位？除了substr 还有没有别的方法？

将字符串转换成数组，然后通过数组的下标去除最后两位，然后通过concat\_ws再还原成字符串。

#### hive 中一个分区表怎么删除其中一个分区？

alter table 表名 drop partition (分区字段=分区值);

#### 两张十万数据的表，进行左连接后发现数据膨胀会怎么处理

先对数据进行筛选，然后再进行表格的连接；

或者将数据计算之后的结果存储到临时表中，然后用临时的中间表进行表格的连接；

或者对两个表格进行分桶。

#### 如何查看执行计划

使用 explain 来查看sql的执行计划

#### 优化器的是如何进行优化的

优化器是强制的修改表格sql语句的运行逻辑，然后在某种情况下可以达到优化的效果。

#### sql语句的去重

insert overwrite table 表名 select distinct \* from 表格；

insert overwrite table 表名 select 列名 from 表格 group by 列；

insert overwrite table 表名 select 列名 from (select 表名.\*,row\_number() over(partition by 列 order by 1) r from 表) where r=1;

#### **hive怎么更新数据效率更高**

最高的方法，使用hadoop fs将数据下载下来，然后使用Linunx命令更新之后再上传。

#### hive的sql语句是怎么运行的

hive窗口中编写sql语句，然后会去Mysql里面进行表格结构和位置的查询，如果表格结构没有问题，那么就会将sql语句拆分成不同的关键字，然后调用mapreduce对应关键字的java脚本模板，如果mr的模板文件运行没有问题，就会调用yarn进行资源的申请和分配，然后使用mr进行数据的计算，最后将计算的结果再回显到hive数据的窗口中。

#### hive中的数据插入流程是怎么样的

就是hdfs数据写入的流程

#### 除了存储过程还有没有其他的？

函数、触发器、包

#### 什么情况下会用到触发器

前置触发器：在数据写入到表格之前，先对写入的数据进行检查，查看数据是否符合写入表格的要求

后置触发器：当数据已经写入到表格了，对表格的数据进行同步更新或者是表格操作日志的记录等等

#### 多表查询是不是一定要用嵌套，还有什么情况下可以考虑不用嵌套

除了嵌套，还有表联合查询，exists()进行多表查询也可以

#### 怎么提取数据，数据抽取方式

全量抽取 在小表或者是维度表格获取是计算后的表格，先对表格进行truncate，然后再全量插入

增量抽取 我们项目组里面，一般是以天为单位，进行日期分区的增量抽取

镜像对比的抽取 一般用来历史记录上进行数据的对比更新或者数据的补录等等

#### 小文件产生的原因

* 每次insert插入产生小文件
* 分桶产生小文件
* 系统本身产生的文件。比如每天的财务数据，数据量不是特别大

#### 小文件处理方法

在map执行前合并小文件，减少map数：

CombineHiveInputFormat 具有对小文件进行合并的功能（系统默认的格式）

set mapred.max.split.size=112345600;

set mapred.min.split.size.per.node=112345600;

set mapred.min.split.size.per.rack=112345600;

set hive.input.format=org.apache.hadoop.hive.ql.io.CombineHiveInputFormat;

这个参数表示执行前进行小文件合并，前面三个参数确定合并文件块的大小，大于文件块大小128m的，按照128m来分隔，小于128m，大于100m的，按照100m来分隔，把那些小于100m的（包括小文件和分隔大文件剩下的），进行合并。

#### 数据倾斜的业务场景

**（1）空值产生的数据倾斜**

**场景说明**

在日志中，常会有信息丢失的问题，比如日志中的 user\_id，如果取其中的 user\_id 和用户表中的 user\_id 相关联，就会碰到数据倾斜的问题。

**解决方案**

**解决方案 1：user\_id 为空的不参与关联**

select \* from log a join user b on a.user\_id is not null and a.user\_id = b.user\_idunion allselect \* from log c where c.user\_id is null;

**解决方案 2：赋予空值新的 key 值**

select \* from log a left outer join user b on case when a.user\_id is null then concat('hive',rand()) else a.user\_id end = b.user\_id

**总结**

方法 2 比方法 1 效率更好，不但 IO 少了，而且作业数也少了，方案 1 中，log 表 读了两次，jobs 肯定是 2，而方案 2 是 1。这个优化适合无效 id（比如-99，’’，null）产生的数据倾斜，把空值的 key 变成一个字符串加上一个随机数，就能把造成数据倾斜的 数据分到不同的 reduce 上解决数据倾斜的问题。

改变之处：使本身为 null 的所有记录不会拥挤在同一个 reduceTask 了，会由于有替代的 随机字符串值，而分散到了多个 reduceTask 中了，由于 null 值关联不上，处理后并不影响最终结果。

**（2）不同数据类型关联产生数据倾斜**

场景说明

用户表中 user\_id 字段为 int，log 表中 user\_id 为既有 string 也有 int 的类型， 当按照两个表的 user\_id 进行 join 操作的时候，默认的 hash 操作会按照 int 类型的 id 进 行分配，这样就会导致所有的 string 类型的 id 就被分到同一个 reducer 当中

**解决方案**

把数字类型 id 转换成 string 类型的 id

select \* from user a left outer join log b on b.user\_id = cast(a.user\_id as string)

#### hive中怎么给数据打标签

使用分析函数进行数据的统计，然后可能需要对数据case when 进行判断。

#### 用union all合并两个表，新表中怎么查询某个数据原本是什么表，用什么语句

select \* from

(select a.\*,sysdate,'a' from a

union all

select b.\*,sysdate,'b' from b);

#### 有个文件需要周一早上8点运行，用linux语句怎么运行？（不能借助kettle）

crontab

0 8 \* \* 1

#### 把一个列变成一个数组都有什么函数

collect\_list(列名) collect\_set(列名)—去重

#### 降序排序空值排在前面还是后面？

最前 (空值默认最大)

可以在最后加nulls last/first 改变空值顺序

#### hive SQL和SQL的区别

Hive-sql不支持等值连接，而sql支持；

Hive-sql不支持“Insert into 表 Values()”、UPDATA、DELETE操作，而sql支持；

Hive-sql不支持事务，而sql支持。

#### hive中order by,sort by, distribute by, cluster by作用以及用法；

cluster by 列名：对这个列进行升序排序

order by 列名 asc|desc：对这个列进行升序或者降序的排序

distribute by 列名 [ sort by 列名 asc|desc ]：相当于对这个列进行了一个分区的操作，如果要实现排序功能，需要和sort by 结合使用。

order by是全局排序，sort by是组内排序。distribute by sort by可以结合桶表使用，给桶中的数据排序。

#### 数据清洗

缺失值清洗：确定范围，去除不需要的字段，填充缺失内容，重新取数

格式内容清洗：统一格式，去除多余字符，校验字段内容

逻辑错误清洗：去重、去除不合理值、修正矛盾内容

非需求数据清洗：删除不需要的字段

# 报表部分

#### finereport的报表设计流程

首先要知道报表的输出形式，要知道报表的数据来源，要知道数据的计算方法，要知道数据的筛选条件...

定义报表是普通、聚合、决策，然后定义数据的模板，定义模板参数、设置可视化的图表



#### 做报表开发，如何确保你拿到的数据时准确的

我们都是根据需求来验证的，如果计算的话，都有计算公式，我会自己先算一遍，然后通过调用函数对比计算结果。

如果是存储过程的话，我会根据需求中的取值逻辑和判断，验证一下取值是否正确

#### 你们用的报表工具是收费的吗？都少钱？

这方面我不太清除，都是公司统一采购的，价格方面我没有关注

#### 做出的报表是以什么文件或者形式发给客户，客户又是通过什么方式查看报表的呢？

首先要看是什么报表，没张报表都有url连接，如果是帆软或者cognos是有客户端的，以web形式展现的，客户有登入账户密码的，我们也有，但是权限不一样，我们相当于系统管理员，是可以操作报表系统的，我们把报表文件，帆软是.cpt文件格式，上传到服务器，然后我们自己把文件放到对应的菜单目录下，自己运行下看是正常可以的，那么客户端那边也可以看到。

#### 帆软和tableau的区别（传统bi工具和商务智能bi工具的区别）

帆软是oltp联机事务处理产品，定位于企业数据采集，满足企业个性化报表需求，同事具备强悍的列印和填表等功能，能制作复杂的中国式报表，但是数据分析能较弱。

tableau是olap联机分析处理产品，具有很强的数据分析能力，能够灵活的在前段进行数据分析操作，无需编写代码，缺点是无法满足企业复杂报表。

#### 假如开发了一张报表，下午三点才出，早9点就要看到数据，你会去做什么样的评估才能给到这样的答复

先看这个需求的复杂程度，以及这个需求涉及到的面是否广；

这个需求之前对应的模块和功能是否开发过；

这个需求是否是一个要从头开发的需求，还是一个对原有功能做修改的需求。

# ETL部分

#### 什么是映射

在etl过程中，源表和目标表的对应关系就是一种映射关系

在逻辑模型中，表和表之间的关系也叫映射。

mapping = 映射

#### 什么是etl

etl，英文名称是extract—transform—load的缩写，用来描述将数据从来源端经过抽取，转换，加载至目标端的过程。

#### kettle抽数一般遇到什么问题，你们是怎么解决的？

kettle在不同的数据库抽取数据时，有时会出现中文乱码问题

解决方法：如果数据设置正确仍然中文乱码，则可能是因为有的客户端数据库默认的编码集不是utf-8，我们只需要在输出时对数据库行编码设置

如果还不行，可以打开options,添加参数characterencoding,设置值为gbk/utf8

#### 如果抽取的数据有重复，怎么解决

在kettle的核心组件里，有一个字段选择，里面有一个 去除重复记录（uniq rows（hashset））的控件，在抽数的时候，可以通过这个控件对数据进行去重。

具体操作：创建几个核心对象，分别是输入—>表输入，将数据根据字段排序，数据去重，将去重好的数据输出到另一个表中，这里需要注意，数据去重前必须要根据相关字段排序

#### etl脚本开发以后，怎么运维

我们会定期巡检，一般2周做一次检查，检查etl脚本跑数是否正常，是否报错，分析脚本跑数的时间，是否越跑越慢，在这种情况下，需要分析脚本和目标表，比如目标表是否后来建了索引，导致更新表的时候越来越慢，索引会影响更新表的效率。如果是这种情况，可以先删除目标表的索引，更新完了以后在重新创建索引，这样可以提高跑数效率

#### 一张特别大的表，几千万，几亿 的表，怎么通过etl工具同步

在etl工具中，我们插入数据的时候，可以设置批量提交，比如10万记录提交一次，而不是一次性提交。

另外如果特别慢的话，还要考虑是否锁表了，比如对目标表进行更新的时候，转换的状态一直不变，可能是目标表产生了排它锁，导致无法更新，这个要具体分析一下。

#### 数据同步以后，源系统的表结构发生了改变，比如源系统的表增加了字段，你的kettle脚本是否会报错？

不会。因为我们抽数的时候，对源表和目标表的每个字段都做过映射，所以源系统的表增加了字段，我们的脚本是不会报错的。

但是如果这些新增的列，也需要同步到数仓，那我们会对ods层表的结构进行变更，然后修改kettle的脚本，然后重新跑数。

#### 有一张表下午5点才出数据，依赖的表早上十点就已经跑完了，是什么原因导致表这么慢出数据（从数仓同步数据，本来早上9点就要出结果，结果一直到下午4点才出结果）

首先我会看一下调度日志，再次检查一遍是否所有的前置job都跑完了，排除因某一个前置job没跑完，导致目标表的job一直在等待。

排除上面的问题后，在分析是否是性能问题，检查监控日志，看下跑数时间，如果确实是目标表更新慢，在进一步分析慢的原因。比如：检查是否是在目标表上建了过多的索引导致更细慢，

另外，我们平时也要定期（每周或者每2周对之前的跑数job进行检查）检查job是不是越来越慢，可能因为数据量过大导致的，数据量过大的原因可以通过创建索引或者通过分区表的方式进行优化。

在分析问题的过程中我们要关注sql的执行计划，必要的时候也可以使用优化器进行sql优化

#### 有一个job前一天执行成功了，第二天再执行失败了，请分析失败的原因

一种是报错，一种是不报错，但是没有结果

报错，可以查看日志，分析错误原因。如果报主键冲突。说明更新的sql逻辑有问题，我们可以查下代码，更改原来的同步逻辑即可

不报错，看调度任务是否执行，日志中如果有开始，但是没有完成时间，说明一直在等待，这时候需要查询一下是否锁表。我会查下锁表的进程，然后杀死该进程。

#### kettle工具的运用

kettle主要是用到了转换和作业两个不同的功能，使用转换进行ETL数据的处理，主要是处理表格或者数据等结构化的数据，或者接口这样的半结构化数据等，然后对数据进行格式的统一，和数据的有效筛选或者是数据的计算等等，然后将数据增量或者全量的保存到数据表格中。

使用作业主要是进行脚本的定时任务的操作等，作业我们是在windows上进行编辑，然后上传到kettle的linux服务器，使用crontab结合kettle的指令进行数据的定时调度。

#### kettle和sqoop哪个效率高

sqoop操作难，效率高

kettle操作简单，效率低

#### 数据迁移 oracle ->hive

定义数据类型规范：比如oracle（varchar2） -> hive（string）

在hive中重构表（保留核心字段，删除不要的字段，加新字段）

sqoop抽取表，检查表数据是否正确

存储过程，自定义函数，用SQL脚本，shell脚本，pthon，java重新写一遍

#### sqoop和kettle的区别

Sqoop是一款开源的工具，主要用于在Hadoop与传统的数据库间进行数据的传递，可以将一个关系型数据库。

Kettle是一款国外开源的ETL工具，纯java编写，可以在Window、Linux、Unix上运行，数据抽取高效稳定。

#### etl如何保证数据准确性.

统计源表和目标的数据量，看是否一致

抽样，手动计算结果和sql运算结果是否一致

两张表的关键字段做minus计算

如果是hive中，因为hive没有主键约束，可以通过group by 分组 having>=2 过滤，看是否有重复数据

#### sqoop抽取异常有哪些

**1.指定了\n为sqoop导入的换行符，mysql的某个string字段的值如果包含了\n, 则会导致sqoop导入多出一行记录**

sqoop导入时删除string类型字段的特殊字符

-hive-drop-import-delims Drops \n, \r, and \01 from string fields when importing to Hive

**2.sqoop导入数据时间日期类型错误**

hive只支持timestamp类型，而mysql中的日期类型是datetime， 当datetime的值为0000-00-00 00:00:00的时候，sqoop import成功，但是在hive中执行select语句查询该字段的时候报错。

**解决方法是在创建hive表时用string字段类型。**

·

**3.sqoop 从mysql导入hive的字段名称问题**

hive中有些关键字限制，因此有些字段名称在mysql中可用，但是到了hive就不行。

比如order必须改成order1, 下面列出了我们发现的一些不能在hive中使用的字段名称

order => order1

sort => sort1

reduce => reduce1

cast => cast1

directory => directory1

# 建模部分

#### 建模的理解

会使用图形化的界面来表示表格和表格之间的关系，以及表格本身的内容。这个表示数据之间的表达关系的图，就叫做ER图。

怎么去设计某个功能需要的表格，以及中间的小表，每个表格有哪些不同的关系等等，这个就是数据建模的过程

#### 知道表结构吗？有设计过表吗？

**（1）确定表名。**表名要确保其唯一性，表的名称要与用途相符，简略、直观、见名知意。

**（2）确定字段名称。**字段名长度小于64个字符；字段名可以包括字母、汉字、数字、空格和其他字符；字段名不可以包括句号(。)、感叹号(!)、方括号([])和重音符号(、)；字段名不可以以先导空格开头。

**（3）确定字段类型。**设计数据类型，满足字段的不同需要。

**（4）确定字段属性。**如字段大小、格式、默认值、必填字段、有效性规则、有效性文本和索引等。

（5）确定表中唯一能识别记录的主关键字段（主键）

#### 数据建模的思路

1. 先要分析当前的功能模块的核心功能，以及这个核心功能和其他模块之间的业务关系，分析他们的业务是哪种业务模型（1-1 1-多 多-1 多-多）

2. 将他们的关系，编辑到业务模型中

3. 将业务模型转换成逻辑模型，查看在转换的过程中，中间还需要哪些不同的业务表格进行数据的存储和中转

4. 给中间表格进行其他的必要字段的添加

5. 给添加完字段信息的逻辑模型，转换成物理模型，最后再查看下有没有需要修改的字段和名字等等

6. 直接在工具中，生成对应的sql语句，在数据库中运行就可以了

#### 三范式

第一范式：原子性，列不可在分

第二范式：在第一范式的基础上，每个非主键必须完全依赖于主键，而不能依赖于其他列

第三范式：在第二范式的基础上，每个非主键必学直接依赖主键，不能产生传递依赖

#### 维度表事实表

事实表就是你要关注的内容，它是用来存储主题的主干内容的，事实表一般是没有主键的。数据的质量完全由业务系统来把握。

维度表就是观察事物的角度，维度表一般是有主键的。

#### 你们怎么考虑建模

先添加一个CDM，进行表格核心字段的编写，在每个表格上面，去表达他们的业务关系和逻辑，连接完成后，将当前的业务模型，转换成逻辑模型，给自动出现的中间的业务小表，添加其他必要字段信息，最后转换成物理模型。

#### 星型模型跟雪花模型的区别

星型模型：所有的维度表都是和事实表直接相连的时候

雪花模型：多个维度表没有直接和事实表相连，而是通过其它的维度表，间接的连接在事实表

**各自优缺点**

雪花模型的优点：雪花模型的优点是减少了数据冗余

雪花模型缺点：在雪花模型需要事实表和维度表之间的连接较多，因此查询性能会相对较低

星型模型的优点：很多的统计情况下，不需要和外表关联进行查询和数据分析，因此效率相对较高。

星型模型的缺点：数据冗余

#### 什么是代理主键和业务主键？

代理主键是纯数字为了记录数据的变量情况，而业务主键是基于业务系统，可能是数字也有可能是字符，主要是为了确保唯一的记录。

比如 订单编号 对应不同的状态，那么 订单编号 就是业务主键 无论订单状态如何变化，订单编号都是不变的。

在数仓中为了记录订单每次变化的情况，我们可以通过代理主键（可以是自从生产的一个序列的 序号 纯数字）来记录每次状态的变化情况。

这一点跟业务系统的业务主键不一样，通过代理主键作为事实表的外键，对事实表的字段分类（维度，度量）可以做到比较清晰

# 项目细节

#### 一张千万数据量的表和一张抽取的新增的8万数据量的表，在不同的层里，怎么合并两张表？用sql方法？

如果千万的表格有分区，那么直接读取数据全量写入到对应的例如今天的分区中；如果是个普通的表格，那么可以使用insert into table进行数据的追加 select \* from 库名.表名

#### 数据漂移

**1.1 定义**

源数据抽取到ods层中，同一个业务日期数据中包含前一天或者后一天凌晨附近的数据或者丢失当天的变更数据。

**1.2 数据漂移出现的原因**

通常落地数仓的ODS表会按时间切分做分区存储，实际上往往由于时间戳字段的准确性问题导致发生数据漂移。通常有四类时间戳：

modified\_time:数据库记录某条数据更新的时间。

log\_time:数据库日志记录某条数据更新的时间。

proc\_time:具体业务过程发生时间。

extract\_time:数据记录被抽取时间。

1）同一条记录的数据抽取时间extract\_time明显是晚于另外三个时间的，如果用这个字段切分，ODS某个分区中的数据会包含前一天末尾的数据，并丢失当天末尾的数据。

2）如果用数据库记录的更新时间modified\_time，前台业务系统手工订正数据时可能会遗忘同步更新该时间，导致该抽取的数据被遗漏掉。

3）另外，由于网络或者系统压力问题，log\_time或者modified\_time可能会晚于proc\_time，导致数据漂移。

4）如果我们直接使用proc\_time时间进行切分，这种情况仅仅对包含一个业务过程的ODS表有效果，如果该表每条记录需要存储多个业务过程，则用proc\_time切分会丢失其他发生在当天的业务过程记录。

**1.3 处理数据漂移的方式**

1.多获取后一天的数据

既然很难解决数据漂移的问题，那么就在ODS每个时间分区中向前、向后多冗余一些数据，保证数据只会多不会少，而具体的数据切分让下游根据自身不同的业务场景用不同的业务时间proc\_time来限制。但是这种方式会有一些数据误差，例如一个订单是当天支付的，但是第二天凌晨申请退款关闭了该订单，那么这条记录的订单状态会被更新，下游在统计支付订单状态时会出现错误。

2.通过多个时间戳字段限制时间来获取相对准确的数据

1) 首先根据log-time分别冗余前一天最后15分钟的数据和后一天凌晨开始15分钟数据，并用modified\_time过滤非当天数据，确保数据不会因为系统问题而被遗漏

2) 然后根据log\_time获取后一天15分钟的数据；针对此数据按照主键根据log\_time做升序排列去重，因为我们要获取的是最接近当天记录变化的数据（数据库日志将保留所有变化的数据，但是落地到ODS表的是根据主键去重获取的最后状态的数据）

3) 最后将前两步结果数据做全外连接，通过限制业务时间proc\_time来获取我们所需要的数据.

#### 你们的项目组表右多少张，数据量大概是多少

表大概有300张左右，数据量在百亿级别

#### 每天的数据量有多少。最大的表有多大？

50-60g（5-6千万行） 3-4亿行（几百/几千亿行）

#### 什么时候用到存储过程

oracle的存储过程一般是在数据的ETL过程中使用的，或者对sql语句进行优化，或者是编写一个固定功能的时候使用存储过程，然后根据存储过程的逻辑决定是否要输入或者输出的参数，如果没有输出的参数可以使用call调用，有输出的Out参数就要放在代码块中进行调用了。

#### 你在工作中遇到过哪些报错，什么原因导致的

存储过程里遇到异常，主键冲突，内存溢出，除数为零，系统等待某一个资源，时间超时

#### 查询两千万的数据要多久

要看取数规则，限制条件的多少。关联表少的时候很快，几分钟就出来了。条件复杂的，关联条件多的话，要跑半个小时左右。

#### 标签，字段，口径是同一个东西

#### 什么是维度退化

比如天维度退化到星期维度

#### 项目的粒度

粒度就是同一维度下，数据统计的粗细程度，比如 天维度相对于月维度来说，粒度就比较细。

#### 每天同步的表有哪些，增量数据有哪些？

一般事实表都是增量的。比如借记卡的开户信息表，贷记卡合同信息表，交易信息表，存款信息表等。

每天同步新开户，新合同，新的交易流水，新存款等。

#### 几十亿的数据你们怎么同步？每天更新的数据量有多大？

增量同步，对于交易明细表每天有几千万的增量

#### 取数部分是按什么取的，怎么保证抓的数据尽量少又不漏？

时间戳区来取的，我们是取最近7天的数据作为增量

（华为的做法）我们是根据会计期进行取数并将数据同步到下游的目标表，一个会计期的时间是1个月

**或者说 ：**时间戳T+1的方式。今天同步昨天的数据

#### 项目数据的来源是什么，需求是怎么来的，拿到需求以后怎么做的？

数据来源于业务系统，包括个人贷款系统，公司贷款系统，票据系统，总账系统，基金理财系统等。

需求是客户先提到需求分析那里，然后需求分析师会对需求进行拆解和澄清，确认取数逻辑，然后我们会开会讨论如何实现，包括可行性分析，设计好模型表，然后我们进行开发。

#### 存储过程写的多吗，一个版本写多少个，一个有多少行，写过最复杂的存储过程是什么，存储过程中怎么记录日志，找出异常？

嗯，我主要就是写存储过程的。

一个版本大概要写2,3个，每个7,8百行，

最复杂的是统计交易明细表，因为粒度比较细，涉及到的表非常多，还要对交易类型和机构进行过滤等。写了1000多行代码。

存储过程每个步骤都会记录日志，也会记录异常

#### 说说项目中做过的指标有哪些？用到的表有哪些？

客户余额，日均余额统计。表有：客户交易明细和客户信息、币种、机构、贷款合同等表

存款账户交易月汇总指标（根据存款账户不同的交易类型来统计交易笔数和交易金额）。表有：客户交易明细和客户信息、币种、机构等表。

#### dm层如何保证数据质量，如何保证数据时正确的？

* 首先在开发阶段，我会仔细分析并理解需求，按照指标计算的公式完成进行开发。在这个阶段，如果我认为需求有问题或者我对需求有疑问会跟我们的产品经理进行沟通和确认
* 开发以后，先进行单元测试，单元测试的时候我会先按照计算公式进行手工计算，然后通过代码再算一遍，对比两次计算结果，如果计算结果一致说明没有问题。
* 之后，我们还有一个代码评审环节，可以进一步确认我们对需求的理解是否正确。
* 进入测试阶段以后我们团的测试会进行充分的测试，对于他们发现的问题我会分析问题并解决，然后对发现的有效缺陷进行分析，看看是否在其他地方也有类似的问题，避免测试漏测，也可以提高我的代码质量。
* 上线以后，对于我们团队的bug，我会进行分析，总结经验，避免以后再次发生类似的bug。

#### 怎么设计表的，怎么设计mapping的，表是谁设计的

这个主要是设计人员设计都，设计好后目标表的字段以及来源于哪些表

#### 存储过程如果出现异常怎么办

在plsql中使用exception，在exception中把过程数据记录到错误日志表，然后对主过程的事务做回滚，并且针对异常我们会记录异常的日志，方便后续问题的分析

#### 如果有10张表，关联以后，会有性能问题，问怎么优化

可以分别每5张表进行关联，产生2个中间表，在把2个中间表进行关联

#### 一个plsql脚本很慢，如何分析到底哪里慢

通过日志，记录每步耗时，分析到底哪一步慢，

然后找到最慢的sql部分，看到底是查询慢还是更新慢，

如果查询慢，看select后面是否调用了其他的自定义函数，注释函数，再查性能是否提升，如果调了自定义函数之后变慢，则需要优化自定义函数的性能

如果仍然很慢，这个时候就要分析数据量的变化和执行计划，主要看数据量的变化情况，是否突然数据增量比较快，导致数据量暴增；还可以通过日志看，脚本执行的效率是否越来越慢，又分为数据量变得越来越大 或者 有人改过代码。

数量多了，之前同步的方式可能需要更改

如果有人为了优化性能，建了更多的索引导致 插入数据越来越慢， 这时候需要平衡查询和更细的性能，有可能需要增加服务器的配置

#### 项目难点

主要难点是在业务的理解上。通过看文档，问同事的方法解决业务难点

#### a表是主表，80万数据，b表是扩展表，100万数据，这两张表相同主键是学生编号，根据这个场景，判断是否存在异常？什么原因造成的数据重复

a表作为主表，80万数据，通常外关联才有主表和从表之分，如果是外关联，那么a表作为主表的数据肯定是全部取的，但是关联以后的结果集比如b表的数据变成了100万，明显数据量多了20万，属于数据发散。通常数据发散的原因可能是关联条件有问题，比如少些某个关联字段导致查询出来的结果集变大，所以我认为应该重新检查和分析一下a表和其他表的关联条件是否遗留某个关联字段。

还有一种可能，a表的数据刚开始就有一部分重复数据，比如原来是80万，但是没有按照三范式的原则来创建主键，结果更新的时候插入了部分重复记录，导致结果集发散。这种情况可以先对主表数据按照主键字段去重，然后再表中创建主键 或者 对关联字段创建唯一索引，通过唯一约束来避免此问题