1 代理主键 和 业务主键

代理主键是纯数字为了 唯一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 纯数字的） 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 ， 通代理主键还作为事实表的外键，对事实表的字段分类(维度，度量)可以做到比较清晰。

2 去重的方式 ，表中 id 不同但是其他的数据都一样

找出我们判断是否重复的 列 （一列或者 多列） ，然后按照这列分组 ，找出重复记录 （ count(\*) >= 2） 然后删除重复记录 。

如果要说伪列 rowid 一定说一下我之前用的是 oracle数据库 ，将之前也要跟他确认一下 你们用的是什么数据库 。

找出重复的记录

SELECT \*

FROM TABLE\_1

WHERE T.要去重的列 IN ( SELECT 要去重的列 FROM TABLE\_1 GROUP BY 要去重的列 HAVING COUNT(\*) >= 2 )

删除

delete

from TABLE\_1 T

WHERE T.要去重的列 IN ( SELECT 要去重的列 FROM TABLE\_1 GROUP BY 要去重的列 HAVING COUNT(\*) >= 2 )

伪列去重是通过物理地址 来去重 ， 先查出 物理地址 唯一的记录 ，然后 重复记录就是 rowid not in ( 唯一记录的结果集 )

DELETE FROM 表

where rowid not in ( 唯一记录的结果集 ) ;

3 三种关联机制（ 执行计划中的关联机制）的优先级

如果是说哪种最好 ，回答 没有最好的，根据不同的情况我们可以指定 走某种关联机制。

基本顺序 排列组合 优于嵌套 ，优于 哈希 但是我们在优化的时候 会先查询 执行计划 。

我们看执行计划主要看什么和 看执行计划的目的：

1 表的关联机制

2 看扫描方式 （全表 或者 索引）

3 消耗 （跟数据量 还有 1 ，2 ）

4 查看执行计划的目的是为了我们后续去更改执行计划。 （在 SELECT 或 INSERT 语句后面加 HINTS , /\*+ 命令 \*/） ；

4 往一个大表中插入数据 怎么提高效率（性能） ？

优先回答 ，往一张大表中插入数据 ，不能在这张大表中创建索引 ，可以插完数据以后再创建索引 ，

如果已经有索引了，可以让索引先失效。

(可以在表上直接 更改索引的状态 让其失效 或者通过命令

drop index 索引名 ;或者

1. alter table xxxxxx move tablespace xxxxxxx 命令后，索引就会失效。

2. alter index index\_name unusable，命令使索引失效。)

5 存储过程的异常处理 怎么插入日志

异常处理时通过 exception 进行判断

语法：

exception

when OTHERS THEN

ROLLBACK ; -- 1 回滚当前事务

v\_ErrorText := sqlcode|| SQLERRM|| '存储过程[Proc\_Update\_ID\_NO]执行失败！'; -- 2 捕获 异常编码 和 异常信息

RAISE\_APPLICATION\_ERROR(-20003,v\_ErrorText);

将异常信息 v\_ErrorText 记录异常日志。

在 exception 的异常判断后面 then 之后调用一下 记录异常日志的存储过程。

记录异常日志以后 ，我们可以插入 日志表 分析PLSQL执行错误的问题。

6 一个大表里面有很多维度，每个维度的评分要大于60分

比如一个表中 有姓名 ，地址 ，性别 等等，我们如何实现这些维度的分数大于 60 分

两种方式：

1 关联实事和维度表 ，然后对这个结果集的维度列 进行 case when 的判断 ， 如果不确定已经有 分数了 那么我们需要写条件判断。

2 通过视图 查出所有积分都大于60的数据 然后用 IN （ 视图中满足条件数据 ）

3 如果有分数了 我们通过 分组聚合 的方式 过滤出 维度的最低分数 > 60 AND 维度的数量 = 全部维度的数量。

7 oracle 9i 和 11G 12C的区别

i是internet的意思，表明当时是internet互联网盛行的年代。

g是grid，网格运算。为了迎合分布式计算而推出的版本。

c是cloud，云计算的意思。

还有每次数据库版本升级 会新出一些新的函数 。

8 oracle 如何 分页

可以通过 伪列 ROWNUM

查出某几行的数据 。

比如 rownum rn ,

rn <= 100 这个就是查前100行 ，相当于 第一页

或者 rn between 101 and 200 这个相当于第二页 101~200 行。

9 电商订单状态变化如何保存所有的状态 和 最后成交的订单记录。

针对这种历史记录要保存 统一都是拉链表 ，也叫做 缓慢变化维。

在拉链表中 最新记录有一个标记字段 比如 1 有效 0 无效， 他们用 update 进行逻辑删除。

如果要删除所有无效的记录 ，那么可以 DELETE 进行物理删除 加上条件 WHERE 标记字段 = 0

10 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 也叫做 映射 。

mapping = 映射。

11 数据 口径 是什么

这个是描述 维度 的粗细的一种表达方式 。

跟粒度差不多 ，比如 统计口径为 ： 月

时间口径/时间粒度 从大到小 年 季度 月 周 天 ，时分秒

12 性能优化 ，如果有10张表 ，关联以后 会有性能你问题，问你怎么优化

可以通过 中间表 进行优化。

比如 A1 ~ A10 ，直接关联有性能问题 ，我们可以分别讲 A1~A5 关联成一张 A1\_5 的中间表 A6~A10 生成第二个 A6\_10的中间表

然后再把2个中间表进行关联。

跟视图不一样，视图不存储数据 ，而中间表 会存储数据 。

13 一个PLSQL脚本很慢如何分析到底哪里慢

通过日志 记录每步的耗时 ，分析到底哪一段最慢。

然后找到最慢的SQL部分 看到底是 查询 还是 更新 慢。

比如查询慢 ，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，则需要优化自定函数的性能。

如果仍然很慢 ，这时候要分析 数据量的变化 和 执行计划 。

数据量的变化 可以通过日志看 是否是越来越慢 ，又分为数据量变的越来越大 或者 有人改过代码 。

1 数量多了 之前的同步方式 可能需要修改 。

2 如果有人为了优话性能 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 更新的性能。有可能需要 增加服务器配置。