1、如何创建表分区

按DEPTNO进行LIST分区

CREATE TABLE MYEMP2

(

EMPNO NUMBER(4) PRIMARY KEY,

ENAME VARCHAR2(10),

JOB VARCHAR2(9),

MGR NUMBER(4),

HIREDATE DATE,

SAL NUMBER(7,2),

COMM NUMBER(7,2),

DEPTNO NUMBER(7,2),

CONSTRAINT EMP1\_FK FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO)

)

PARTITION BY LIST (DEPTNO)

(

PARTITION MYEMP\_DEPTNO\_10 VALUES (10),

PARTITION MYEMP\_DEPTNO\_20 VALUES (20) ,

PARTITION MYEMP\_DEPTNO\_30 VALUES (30) ,

PARTITION MYEMP\_DEPTNO\_40 VALUES (40)

)

2、如何创建唯一索引

语法结构：创建索引

CREATE [UNIQUE] INDEX index\_name ON table\_name(column\_name[,column\_name…])

语法解析：

1）UNIQUE:指定索引列上的值必须是唯一的。称为唯一索引。

2）index\_name：指定索引名。

3）tabl\_name：指定要为哪个表创建索引。

4）column\_name：指定要对哪个列创建索引。我们也可以对多列创建索引；这种索引称为组合索引。

3、oracle 给表授权grant

oracle 给表授权的示例：

grant select,update,delete,insert on TXXX to UXXX;

其中TXXX为表名 UXXX为用户名

如果报错：ORA-01749，请以SYSDBA的权限登陆。Connect as SYSDBA

4、oracle分区索引与全局索引的区别？

分区索引就是在所有每个分区上单独创建索引，它能自动维护，在drop或truncate某个分区时不影响该索引的其他分区索引的使用，也就是索引不会失效，维护起来比较方便，但是在查询性能稍微有点影响。全局索引是在全表上创建索引，它可以创建自己的分区，可以和分区表的分区不一样，也就是它是独立的索引。在drop或truncate某个分区时需要重新建索引；

分区索引在跨分区时，特别是跨N个分区时，就没用了，或者说，即使有用也是用处基本可以忽略不计的，因为在跨分区查询时分区索引效率确实不高。

全局索引则是一直有效的。

那么分区索引好在什么地方，我们举个例子，假设一张表有1000万条记录，分100个区，每取10万条，在同一个字段上有分区索引，和全局索引。而且基本上能保证每次查询都在同一个区。

那么如果调用全局索引，那么你要从1000万条素银信息中找你需要的内容；但是如果你调用分区索引，那么你只要从10万条索引信息中找就可以了。所以查询速度就会加快，这就是分区索引的好处。

举个例子就是，一个人当了某地的导游（假设有100个景点），那么他要掌握的东西一定非常多，我们假设他都记住了，那么他所耗费的精力（内存）一定非常大。假设这个地区有100个导游（每个景区一个，去其他景区基本抓瞎，可是每个景区的导游都是一个名字，假设就叫就叫“a”），每人一个景点，那么对于每一个导游来说，就是闭着眼睛也能记住。那么对于只是逛一个景点，或者时间充裕的逛两三个景点的人来说，找景区导游就够了。

但是如果一个人就要短时间内逛10个景点，每到一个景点还要换导游，那是不是很麻烦，而且因为时间短，为了省事，可能就找地区导游。

5、char和varchar的区别

char的长度是不可变的，而varchar的长度是可变的，也就是说，定义一个char[10]和varchar[10],如果存进去的是‘csdn’,那么char所占的长度依然为10，除了字符‘csdn’外，后面跟六个空格，而varchar就立马把长度变为4了，取数据的时候，char类型的要用trim()去掉多余的空格，而varchar是不需要的。

尽管如此，char的存取数度还是要比varchar要快得多，因为其长度固定，方便程序的存储与查找；但是char也为此付出的是空间的代价，因为其长度固定，所以难免会有多余的空格占位符占据空间，可谓是以空间换取时间效率，而varchar是以空间效率为首位的。

再者，char的存储方式是，对英文字符（ASCII）占用1个字节，对一个汉字占用两个字节；而varchar的存储方式是，对每个英文字符占用2个字节，汉字也占用2个字节。

两者的存储数据都非unicode的字符数据。

6、truncate语句与delete语句区别，事物，with用法

7、

美国征信机构的征信业务中，收集客户信息的主要指标包括：  (1)个人识别信息，包括姓名、生日、现住址、以前的住址等通常的个人信息，除此以外还包括社会保险号、工作、职务以及雇主信息； (2)讨债记录；

(3)公共记录信息，包括个人破产记录、法院诉讼判决记录、税务扣押记录和财产判决记录，以及抵押记录；

(4)个人信用信息，包括信用卡、房屋贷款、分期贷款情况，每个信用账户的帐号、开始日期、信用额度和贷款数量、余额、最近更新日期、最少还款额，以及过去7年的贷款偿还情况；

5)查询记录。包括查询人的名字以及查询目的，包括本人授权银行或信用卡公司进行的查询以及自行索取报告、信用卡公司的售前查询等。 此外，还有一些法院公开宣判的与个人财务有关的信息，但没有税务和公安等其他信息。征信机构在收集信息数据过程中，除了考虑到法律上对各种个人数据使用的约束条件以外，还要比较获取数据的成本和价值。提供信息的来源主要有：银行或贷款机构、保险公司、讨债公司、政府和法庭等。

8、优化结合实例；特别是执行计划（看扫表方式，看表的关联方式，看cpu消耗，看耗时）

看到扫表方式是全表扫描第一时间就应该想到没有加索引，可以从索引的这个方向去优化，看表连接方式

9、看CPU的消耗，

show parameter CPU;

10、几亿的数据的表，怎么插入一列？插入之前需要注意什么？

11、命令查看执行计划；

12、用过拉链表吗？举个例子。

13、全局索引和分区索引；分区索引怎么建？

14、表连接方式的选择，怎么为最优选择；

15、BI理论，分层的意义；

16、高水位的问题怎么处理。

17、星型模型

星型架构是一种非正规化的结构，多维数据集的每一个维度都直接与事实表相连接，不存在渐变维度，所以数据有一定的冗余，如在地域维度表中，存在国家A 省B的城市C以及国家A省B的城市D两条记录，那么国家A和省B的信息分别存储了两次，即存在冗余。

雪花模型

当有一个或多个维表没有直接连接到事实表上，而是通过其他维表连接到事实表上时，其图解就像多个雪花连接在一起，故称雪花模型。雪花模型是对星型模型的扩展。它对星型模型的维表进一步层次化，原有的各维表可能被扩展为小的事实表，形成一些局部的" 层次" 区域，这些被分解的表都连接到主维度表而不是事实表。如图所示，将地域维表又分解为国家，省份，城市等维表。它的优点是：通过最大限度地减少数据存储量以及联合较小的维表来改善查询性能。雪花型结构去除了数据冗余。

星型模型因为数据的冗余所以很多统计查询不需要做外部的连接，因此一般情况下效率比雪花型模型要高。星型结构不用考虑很多正规化的因素，设计与实现都比较简单。雪花型模型由于去除了冗余，有些统计就需要通过表的联接才能产生，所以效率不一定有星型模型高。正规化也是一种比较复杂的过程，相应的数据库结构设计、数据的 ETL、以及后期的维护都要复杂一些。因此在冗余可以接受的前提下，实际运用中星型模型使用更多，也更有效率。

18、什么情况下不会用到三范式建表；

19、hints干预执行计划实现多线程的语法

/\*+ PARALLEL(表名1,并行数)[(表名2,并行数)……] \*/ --指定开启多少个并行|并发（一般为2、4、8……）

20、几亿的数据的表，删除重复行；

21、增量同步的时候，目标表有重复的行，原表数据也有这么一行，还能插入或者更新吗？

会报错。

22、有几个在自我介绍后就开始问你上班打卡的地点，原公司的地点，原公司的业务不包含外包业务的等等；

怀疑自我介绍和项目介绍这块做得太烂，磕磕巴巴说完的；开局就被怀疑了。

23、看你写了好几个银行项目，都做了多久的项目？对指标的了解，对项目整体的情况说一下；--卡死，项目细节准备不足，没那么多给他说了；