|  |  |  |
| --- | --- | --- |
| **آزمون SQL** | | |
| شرایط حین آزمون:  استفاده از کتاب، جزوه یا اینترنت مجاز و مشورت با دیگران غیرمجاز است. | تعداد ســـــوالات: 4  زمان پاسخگویی: 4 ساعت | تاریخ تهــیه: 12/12/1396  طراح سوال: امیرحسین فرمد |

.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | کوئری بنویسید که خروجی آن مطابق شکل زیر است و از منطق زیر پیروی می کند:   * سطر اول روزی را در یک سال اخیر نمایش می دهد که حداکثر تعداد نامه در آن ثبت شده است. * سطر دوم ماهی را در سال جاری نمایش می دهد که حداکثر تعداد نامه در آن ثبت شده است.  |  |  | | --- | --- | | تعداد نامه | زمان | | 115 | 10/12/1395 | | 1125 | آبان | |
| SQL>  درک ساختار union است برای ایجاد ساختار کلی کوئری:  Select ..  Union  Select ..  در هر بخش می بایست گروه بندی بر روی زمان صورت گیرد و درک group by اصل است:  group by substr(to\_char((let\_record\_date),'YYYY/MM/DD ', 'nls\_calendar = persian '),1,10)  سنجش میزان دقت در خواندن مطالب که سطر اول در یک سال اخیر و سطر دوم در سال جاری را خواسته است:  (sysdate - interval '1' year) < let\_record\_date  (21 مارس معادل با اول فرودین است) < let\_record\_date  در سطر دوم برای نوشتن ماه باید یک لایه کوئری را بیرون آورد و از case یا decode استفاده کرد. |
|  | کوئری بنویسید که خروجی آن مطابق شکل زیر است و از منطق زیر پیروی می کند:   * تعداد نامه های ثبت شده توسط مدیر سیستم و زیرمجموعه اش (در چارت سازمانی) که توسط هیچ کس ارجاع نشده است(فرض می کنیم هیچ سمتی نداریم که دو یا تعداد بیش تری شخص به وی اختصاص داده شده باشد).  |  |  |  |  | | --- | --- | --- | --- | | تعداد نامه | نام و نام خانوادگی شخص | نام سمت | شماره سطر | | 11 | علی علوی | مدیر سیستم | 1 | | 13 | حسین حسینی | معاون | 2 | | 110 | معصومه معصومی | کارشناس | 3 | |
| SQL>  درک subquery و شناسایی جدول اصلی (یعنی TBL\_DEPARTMENT) برای ایجاد ساختار کلی کوئری:  select dep.dep\_name,  (… dep.dep\_id) ,  (… dep.dep\_id)  from TBL\_DEPARTMENT dep  توسط در نوشتن کوئری های سلسه مراتبی (hierarchial) به زبان Oracle SQL ارزیابی می گردد.  select dep.dep\_name  from TBL\_DEPARTMENT dep  start with dep.dep\_id = 1  connect by prior dep.dep\_id = dep.dep\_parent\_id ;  اضافه کردن شماره سطر و نام شخص (corelated subquery):  select rownum,  dep.dep\_name,  (select max (p.per\_first\_name ||' '||p.per\_last\_name)  from TBL\_PERSONS p ,  TBL\_PERSON\_DEPARTMENT pd  where pd.prp\_per\_id = p.per\_id  and pd.prp\_dep\_id = dep.dep\_id) person\_name  from TBL\_DEPARTMENT dep  start with dep.dep\_id = 1  connect by prior dep.dep\_id = dep.dep\_parent\_id ;  اضافه کردن تعداد نامه های ثبت شده و ارجاع نخورده(corelated subquery)  استفاده از not in و توجه به قانون مهم not in که باید null را از آن خارج کرد چرا که در جدول ارجاع هم نامه داریم و هم ایمیل.  استفاده از not exist.  کوئری نهایی:  select rownum,  dep.dep\_name,  (select max (p.per\_first\_name ||' '||p.per\_last\_name)  from TBL\_PERSONS p ,  TBL\_PERSON\_DEPARTMENT pd  where pd.prp\_per\_id = p.per\_id  and pd.prp\_dep\_id = dep.dep\_id) person\_name,  (select count(let.let\_id)  from TBL\_LETTER let  where let.let\_registrar\_dep\_id = dep.dep\_id  and let.let\_id not in (select dlv.dlv\_let\_id  from TBL\_DELIVER dlv  where dlv.dlv\_let\_id is not null)) let\_no\_dlv\_count  from TBL\_DEPARTMENT dep  start with dep.dep\_id = 1  connect by prior dep.dep\_id = dep.dep\_parent\_id ; |
|  | کوئری بنویسید که رشته ای همچون الف را از ورودی به عنوان پارامتر گرفته و عبارت های "BAD" داخل پرانتز آن را تبدیل به "GOOD" کند و در نهایت رشته ای همچون ب را به عنوان خروجی نمایش دهد:   |  |  | | --- | --- | | "DEVIL IS BAD (ANGEL IS BAD ANGEL IS BAD) DEVIL IS BAD" | الف | | "DEVIL IS BAD (ANGEL IS GOOD ANGEL IS GOOD) DEVIL IS BAD" | ب | |
| SQL>  دریافت یک رشته به عنوان پارامتر:  select &P\_INPUT\_STR  from dual;  یافتن محل پرانتزها با instr و درک پارامتر -1 به معنای از انتهای رشته.  برش رشته با Substr و درک دو پارامتره بودن یا سه پارامتره بودن این تابع.  جایگزین کردن با replace.  چسباندن مجدد رشته با concat یا pipe.  select '&P\_INPUT\_STR' ,  ---- find parenthesis  instr ( '&P\_INPUT\_STR' , '(' , 1) open\_parenthesis ,  instr ( '&P\_INPUT\_STR' , ')' , -1) close\_parenthesis ,  ---- separate sentence  substr ( '&P\_INPUT\_STR' ,  1,  instr ( '&P\_INPUT\_STR' , '(' , 1)  ) str\_before\_parenthesis ,  --  substr ( '&P\_INPUT\_STR' ,  instr ( '&P\_INPUT\_STR' , '(' , 1) + 1,  instr ( '&P\_INPUT\_STR' , ')' , -1) - instr ( '&P\_INPUT\_STR' , '(' , 1) - 1  ) str\_between\_parenthesis,  --  substr ( '&P\_INPUT\_STR' ,  instr ( '&P\_INPUT\_STR' , ')' , -1)  ) str\_after\_parenthesis ,  ---- rapace  replace ( substr ( '&P\_INPUT\_STR' ,  instr ( '&P\_INPUT\_STR' , '(' , 1) + 1,  instr ( '&P\_INPUT\_STR' , ')' , -1) - instr ( '&P\_INPUT\_STR' , '(' , 1) - 1  ) ,  'BAD',  'GOOD'  ) new\_str\_between\_parenthesis ,  ---- concat  substr ( '&P\_INPUT\_STR' ,  1,  instr ( '&P\_INPUT\_STR' , '(' , 1)  )  ||  replace ( substr ( '&P\_INPUT\_STR' ,  instr ( '&P\_INPUT\_STR' , '(' , 1) + 1,  instr ( '&P\_INPUT\_STR' , ')' , -1) - instr ( '&P\_INPUT\_STR' , '(' , 1) - 1  ) ,  'BAD',  'GOOD'  )  ||  substr ( '&P\_INPUT\_STR' ,  instr ( '&P\_INPUT\_STR' , ')' , -1)  )  from dual; |
|  | اسکریپت یا کوئری یا شرح عملکرد خود را برای هر یک از بندهای زیر بنویسید:   1. جدولی با دو ستون "شناسه" و "نام" ایجاد کنید و ستون شناسه را کلید اصلی جدول تعریف کنید. 2. با نام های تصادفی در این جدول ده هزار رکورد وارد کنید.  |  |  | | --- | --- | | نام | شناسه | | John | 1 | | Sara | 2 | | Sara | 3 | | Smith | 4 |  1. کوئری بنویسید که اسامی تکراری و فراوانی آن را نمایش دهد:   (اگر هیچ دیتای تکراری وارد نشده است دستی چند سطر تکراری وارد کنید)   |  |  | | --- | --- | | نام | شناسه | | 1 | John | | 2 | Sara | | 1 | Smith |  1. دستوری بنویسید که از هر نام تکراری تنها یکی را باقی بگذارد و مابقی را حذف نماید.   (یعنی در مثال بالا یکی از SARA ها باید حذف شوند) |
| SQL>  آشنایی با create table و create sequnce  برای پر کردن جدول دانشجو یکی از راه های زیر را پیش می گیرد:   1. استفاده از ابزارهای generate dummy data یا 2. نوشتن برنامه به زبان PL/SQL و استفاده از تابع dbms\_random یا 3. جست و جو در اینترنت و یافتن اسامی انگلیسی یا فارسی از وبسایت هایی که اطلاعات آماری ارایه می کنند و با دستور bulk insert آن ها را وارد می کند   آشنایی با group by یا rank() over  ارزیابی میزان حساسیت در حذف داده ها و نوشتن دستور select قبل از حذف و ایجاد backup با ctas (create table as)  دستور حذف داده های تکراری  --1) create table  create table test\_96 (  ID number,  NAME VARCHAR2(40),  constraint COUNTRY\_C\_ID\_PK primary key (ID)  );  -- create sequence  create sequence seq\_test\_96\_1;  -- create trigger  create or replace trigger tri\_seq\_id  before insert  on test\_96  for each row  begin  :new.id := seq\_test.nextval;  end;    --2) create random data  Begin  For IDS in 1..10000  Loop  INSERT INTO test\_96 (NAME) VALUES ( dbms\_random.string('L', 15));  Commit;  End loop;  End;  commit;  select \* from test\_96  order by id desc;  select name, count(\*)  from test\_96  group by name  order by count(\*) desc;  --3) select count of data  select name, count(\*)  from test\_96  having count(\*)>1  group by name;    --backup  create table test\_96\_bkp as  select \*  from test\_96;  --4) delete dulicate name  select id ,  name ,  rank() over (partition by name order by id) rn  from test\_96  where name in (select name  from test\_96  having count(\*)>1  group by name) ;  --  select tbl1.id , tbl1.name , tbl1.rn  from (  select id ,  name ,  rank() over (partition by name order by id) rn  from test\_96  where name in (select name  from test\_96  having count(\*)>1  group by name)  ) tbl1  where tbl1.rn > 1 ;  --  delete test\_96  where id in (select tbl1.id  from (  select id ,  name ,  rank() over (partition by name order by id) rn  from test\_96  where name in (select name  from test\_96  having count(\*)>1  group by name)  ) tbl1  where tbl1.rn > 1) |