CYPRUS INTERNATIONAL UNIVERSITY

CMPE344 Database Management Systems II

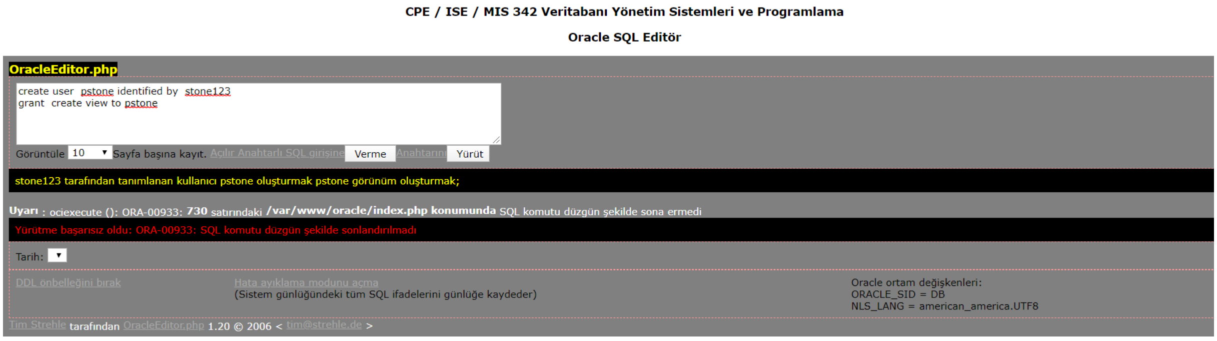
HW3 (Due Date: 13/05/2020) **Barış Karapelit 21707346**

Q1- There is a new employee called Peter Stone hired to the company. Please create an account for him with username “pstone” and password “stone123” and give him a privilege to create “view”.

Answer :

CREATE USER pstone IDENTIFIED BY stone123

GRANT CREATE view TO pstone


CPE / ISE / MIS 342 Veritabanı Yönetim Sistemleri ve Programlama

Oracle SQL Editör

OracleEditor.php

```
create user  pstone identified by  stone123
grant  create view to pstone
```

Görüntüle 10 Sayfa başına kayıt.

stone123 tarafından tanımlanan kullanıcı pstone oluşturmak pstone görünüm oluşturmak;

Uyarı : ociexecute (): ORA-00933: **730** satırındaki **/var/www/oracle/index.php konumunda** SQL komutu düzgün şekilde sona ermedi

Yürütme başarısız oldu: ORA-00933: SQL komutu düzgün şekilde sonlandırılmadı

Tarih:

(Sistem günlüğündeki tüm SQL ifadelerini günlüğe kaydeder)

Oracle ortam değişkenleri:
ORACLE_SID = DB
NLS_LANG = american_america.UTF8

Tim Strehle tarafından OracleEditor.php 1.20 © 2006 < tim@strehle.de >

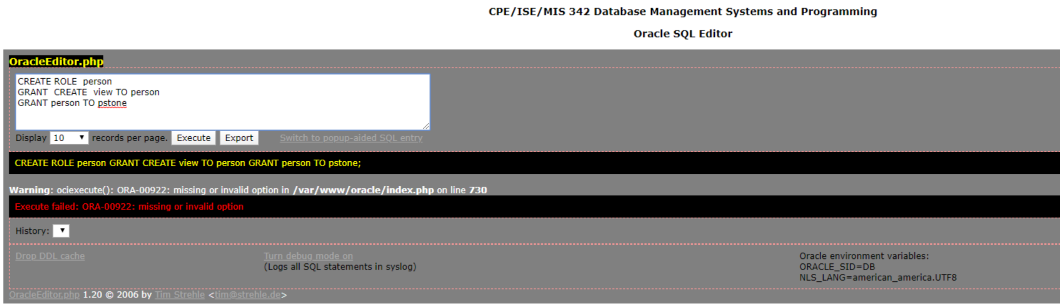
Q2- Explain the purpose of creating a role and assigning a role to a person instead of assigning privileges to that person.

Answer :

CREATE ROLE person

GRANT CREATE view TO person

GRANT person TO pstone


CPE/ISE/MIS 342 Database Management Systems and Programming

Oracle SQL Editor

OracleEditor.php

```
CREATE ROLE  person
GRANT  CREATE  view TO person
GRANT person TO pstone
```

Display 10 records per page.

CREATE ROLE person GRANT CREATE view TO person GRANT person TO pstone;

Warning: ociexecute(): ORA-00922: missing or invalid option in **/var/www/oracle/index.php** on line **730**

Execute failed: ORA-00922: missing or invalid option

History:

(Logs all SQL statements in syslog)

Oracle environment variables:
ORACLE_SID=DB
NLS_LANG=american_america.UTF8

OracleEditor.php 1.20 © 2006 by Tim Strehle <tim@strehle.de>

If we assign privileges to the person, we need to log in to address all of their features one by one, and it's impossible to assign privileges to each one if we think databases are thousands of users. When a new privilege is added, it takes a lot of time for all users to assign, but roles are defined only by a role assignment. So the role is more useful and practical.

Q3- Write an SQL statement to set “state_province” attribute of Locations table as UNUSED.

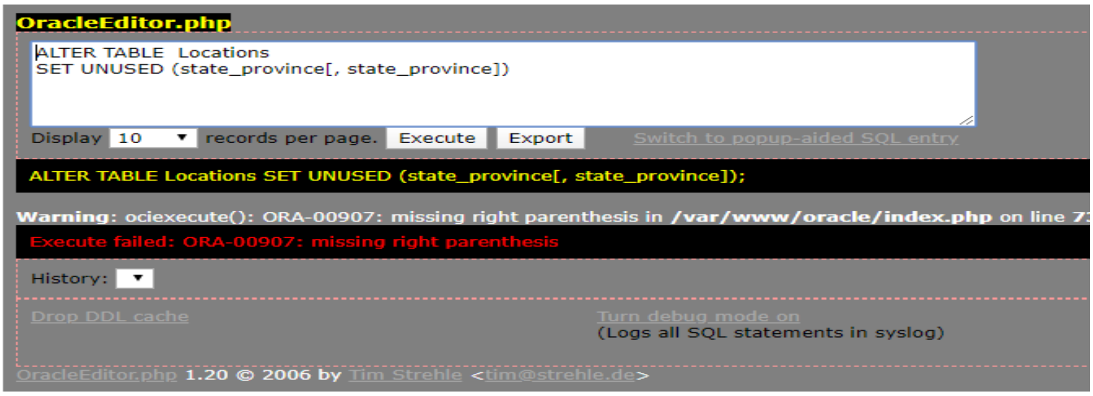
ALTER TABLE system.locations

SET UNUSED (state_province)

OR

ALTER TABLE *system.locations*

SET   UNUSED COLUMN system.*state_provience*


OracleEditor.php
Display 10 records per page. Execute Export
Switch to popup-aided SQL entry
History:
Drop DDL cache
Turn debug mode on
(Logs all SQL statements in syslog)
OracleEditor.php 1.20 © 2006 by Tim Strehle <tim@strehle.de>

```
ALTER TABLE  Locations
SET UNUSED (state_province[, state_province])
```

```
ALTER TABLE Locations SET UNUSED (state_province[, state_province]);
```

```
Warning: ociexecute(): ORA-00907: missing right parenthesis in /var/www/oracle/index.php on line 7
Execute failed: ORA-00907: missing right parenthesis
```

Q4- Write an SQL statement to update the salary attribute in employees table. Add a DEFAULT constraint with system date.

ALTER TABLE system.employees MODIFY salary SET DEFAULT CONSTRAINT

CURRENT_DATE

Q5- Explain the purpose of ON DELETE CASCADE and the difference with ON DELETE SET NULL.

ON GELETE CASCADE also deletes the record in the primary table and automatically deletes records in the sub table where the foreign key is defined, but ON DELL SET NULL is automatically null in the records in the sub table in which the foreign key is defined when you delete the record in the primary table.

OR

ON DELETE CASCADE -> If a parent table is deleted, its child table will be deleted.

ON DELETE SET NULL -> If a parent table is deleted, its child table’s foreign key

referenced this parent table will be deleted.

Q6- What is INDEX. Explain how it works.

Answer:

Index is a method for performance. An index creates entries for values of indexed

columns. By default Oracle uses B-tree indexes.

Lowercase characters in the data can translate them into capitalized columns. In the more active way, we use index in data rankings by calling separately. Data is used to sort from small to large, from old to new date.

-----------------------------------Index use in table--

```
CREATE TABLE NEW_EMP

(employee_id NUMBER(6)
          PRIMARY KEY USING INDEX
         (CREATE INDEX emp_id_idx ON
         NEW_EMP(employee_id)),

first_name  VARCHAR2(20),
last_name   VARCHAR2(25))
```

---------------------------------Index use in out of table--

```
CREATE INDEX upper_dept_name_idx

ON dept2(UPPER(department_name))
```

Q7- Write an SQL statement to insert a new record to Employees table with the following details. Use WITH CHECK OPTION to make sure manager_id=100 is the manager of department_id 60 from Departments table.

employee_id= 141

first_name= Lily

second_name= Bush

manager_id= 100

department_id= 60

Answer:

```
INSERT INTO (SELECT employee_id,first_name, second_name, manager_id, department_id

 FROM system.employees

WHERE employee_id=100 and department_id

IN( SELECT  employee_id

 FROM employees

NATURAL JOIN  departments
WHERE  department_id=60)

WITH CHECK OPTION  )

VALUES(141,'Lily','Bush',100,60)
```

Q8- Write an SQL statement to copy (use INSERT ALL) employee_id, first_name, last_name and commission_pct of all employees into EMP_COMM table and employee_id, first_name, last_name and salary into EMP_SAL table who works in Sales Department.

Answer :

```
INSERT  ALL

WHEN DEPARTMENT =80 THEN

INTO emp_sal VALUES(EMPID,FIRST,LAST,SAL)

WHEN DEPARTMENT != 80 THEN

INTO emp_comm VALUES(EMPID,FIRST,LAST,COMM)

SELECT employee_id EMPID, first_name FIRST,last_name LAST

salary SAL, commission_pct COMM,department_id DEPARTMENT

FROM  employees
```

OR

```
CREATE VIEW EMP_SAL AS SELECT employee_id, first_name, last_name, salary  FROM system.employees

CREATE VIEW EMP_COMM AS SELECT employee_id, first_name, last_name, commission_pct FROM system.employees

INSERT ALL

WHEN DEPARTMENT_ID != 80 INTO EMP_COMM VALUES (employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commision_pct, manager_id, department_id)

WHEN DEPARTMENT_ID = 80 INTO EMP_SAL VALUES(employee_id, first_name, last_name, salary)

SELECT employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commision_pct, manager_id, department_id FROM system.employees
```

Q9- Write an SQL statement to copy (use INSERT ALL) employee_id, first_name, last_name and job_id of all employees into EMP_REP or EMP_ASST tables based on their job_id. If job_id contains “REP” then insert into EMP_REP otherwise if it contains ASST into EMP_ASST.

Answer :

```
INSERT ALL

WHEN JOBID  like ‘%REP%’ THEN

INTO system.emp_rep VALUES(EMPID,FIRST,LAST,JOBID)

WHEN JOBID  like ‘%ASST%’ THEN

INTO system.emp_asst VALUES(EMPID,FIRST,LAST,JOBID)

SELECT system.employee_id EMPID, system.first_name FIRST, system.last_name LAST,
system.job_id  JOBID
```

OR

```
CREATE VIEW EMP_REP AS SELECT employee_id, first_name, last_name, job_id FROM
system.employees CREATE VIEW EMP_ASST AS SELECT employee_id, first_name, last_name, job_id
FROM system.employees INSERT ALL

WHEN JOB_ID LIKE '%REP' THEN INTO EMP_REP VALUES (employee_id, first_name, last_name,
job_id)

WHEN JOB_ID LIKE '%ASST' THEN INTO EMP_ASST VALUES(employee_id, first_name, last_name,
job_id)

SELECT employee_id, first_name,last_name,job_id FROM system.employees
```

OR

```
INSERT ALL

INTO EMP_REP (employee_id, first_name, last_name, job_id) SELECT job_id FROM

System.employees VALUES job_id=”REP”

INTO EMP_ASST (employee_id, first_name, last_name, job_id) SELECT job_id FROM

Employees VALUES job_id=”ASST”

SELECT * FROM system.employees;
```

Q10- Explain how FLASHBACK works.

Flashback: returns the data as it existed at some time in the past . It works by undoing all the changes that were made since that time.

FLASHBACK provides recovering database objects from past without media recovering.

With the FLASHBACK command system, the operations performed in the database can also be undone within the specified time. For example, when data is accidentally deleted from a table, we can restore it with the FLASHBACK command. If there are complete database distortions, we can restore all of them with the FLASHBACK command.