Access Control

Sample database

```
CREATE TABLE Department(
name                VARCHAR2(50),
code                CHAR(5),
total_staff_number NUMBER(2)                NOT NULL,
chair               VARCHAR2(50),
budget              NUMBER(9,1)             NULL,
CONSTRAINT dept_pkey PRIMARY KEY(name),
CONSTRAINT dept_ckey1 UNIQUE(code),
CONSTRAINT dept_ckey2 UNIQUE(chair),
CONSTRAINT dept_check1
CHECK (total_staff_number BETWEEN 1 AND 50)  );
```

```
CREATE TABLE Course(
c#                  CHAR(7),
title               VARCHAR2(200)           NOT NULL,
credits             NUMBER(1)               NOT NULL,
offered_by          VARCHAR2(50)            NULL,
CONSTRAINT course_pkey PRIMARY KEY(c#),
CONSTRAINT course_check1
        CHECK (credits IN (6, 12) ),
CONSTRAINT course_fkey1 FOREIGN KEY(offered_by)
        REFERENCES Department(name) );
```


GRANT statement

GRANT statement is applied by a user to authorise various kinds of access to his/her relational tables/views by another user or class of users

Example

User **scott** would like to grant to user **janusz** read access to a relational table **Department**

```
connect scott/password
GRANT SELECT
ON Department
TO janusz;
```


Example

User **janusz** would like to grant to user **scott** full access to a view **Course_view**

```
connect janusz/password
GRANT ALL
ON Course_view
TO scott;
```


Example

User **scott** would like to grant to all users the rights to delete the rows from a table **Course**

```
connect scott/password
GRANT DELETE
ON Course
TO PUBLIC;
```

Keywords

```
GRANT SELECT ON Department TO janusz;
GRANT ALL ON Course_view TO scott;
GRANT DELETE ON Course TO PUBLIC;
GRANT UPDATE ON qqq TO scott WITH GRANT OPTION;
```


Privileges

```
GRANT SELECT ON Department TO janusz;
GRANT ALL ON Course_view TO scott;
GRANT DELETE ON Course TO PUBLIC;
GRANT UPDATE ON qqq TO scott WITH GRANT OPTION;
```

```
SELECT
```

```
DELETE
```

```
INSERT
```

```
UPDATE <list of columns>
```

```
REFERENCES <list of columns>
```

```
ALL
```


Objects

```
GRANT SELECT ON Department TO janusz;
GRANT ALL ON Course_view TO scott;
GRANT DELETE ON Course TO PUBLIC;
GRANT UPDATE ON qqq TO scott WITH GRANT OPTION;
```

Relational tables

Views

Grantee

```
GRANT SELECT ON Department TO janusz;
GRANT ALL ON Course_view TO scott;
GRANT DELETE ON Course TO PUBLIC;
GRANT UPDATE ON qqq TO scott WITH GRANT OPTION;
```

Users

`PUBLIC`

Granting rights

```
GRANT SELECT ON Department TO janusz;
GRANT ALL ON Course_view TO scott;
GRANT DELETE ON Course TO PUBLIC;
GRANT UPDATE ON qqq TO scott WITH GRANT OPTION;
```

`WITH GRANT OPTION`

Examples

User `scott` grants `SELECT` access rights on `Course` table to user `janusz`

```
GRANT SELECT ON Course TO janusz;
```

User `janusz` selects from `Course` table

```
SELECT *
FROM scott.Course;
```

To eliminate a reference to `scott` user `janusz` creates a synonym

```
CREATE SYNONYM SCourse FOR scott.Course;
SELECT *
FROM SCourse;
```

Examples

User janusz uses a synonym Scourse to selects from Course table

```
SELECT *
FROM SCourse;
```

User janusz grants UPDATE access rights on the columns name, budget from Department table to user scott

```
GRANT UPDATE (name, budget) ON Department TO
scott;
```

User scott updates attribute budget

```
UPDATE janusz.Department
SET budget = budget + 100;
```


Examples

User scott grants ALL access rights to a relational table Student to everyone

```
GRANT ALL ON Student to PUBLIC;
```

Now, anyone can delete a row from Students table

```
DELETE FROM scott.Student
WHERE s# = 6509353;
```


Examples

User scott grants the reference access rights on a relational table Department to janusz

```
GRANT REFERENCES(name) ON Department to janusz;
```

Then, user janusz is allowed to reference column dname when creating a relational table BigDepts

```
CREATE TABLE BigDepts(
dname      VARCHAR(30) NOT NULL,
…          …           …,
)
CONSTRAINT BigDepts_fkey FOREIGN KEY (dname)
     REFERENCES scott.Department(name);
```


Examples

User janusz grants SELECT access rights on all courses offered by Math department to scott

```
CREATE VIEW MathCourse AS (
     SELECT *
     FROM Course
     WHERE offered_by ='Math');
```

```
GRANT SELECT ON MathCourse TO scott;
```

User scott selects the courses offered by Math department

```
SELECT *
FROM janusz.MathCourse;
```


Examples

User janusz grants SELECT with grant option on a relational table Course to scott

```
GRANT SELECT ON Course to scott
WITH GRANT OPTION;
```

User scott is allowed to grant SELECT access rights on a table Course to user joan

```
GRANT SELECT ON janusz.Course TO joan;
```


REVOKE statement

REVOKE statement is applied by a user to revoke the access rights granted to another user or class of users

Example

User scott would like to revoke from user janusz read access to a relational table Department

```
connect scott/password
REVOKE SELECT
ON Department
FROM janusz;
```


Example

User janusz would like to revoke from user scott full access to a view Course_view

```
connect janusz/password
REVOKE ALL
ON Course_view
FROM scott;
```


Example

User scott would like to revoke from all users the rights to delete the rows from a relational table Course

```
connect scott/password
REVOKE DELETE
ON Course
FROM PUBLIC;
```


Example

User janusz revokes SELECT with grant option on Course table from scott

```
REVOKE SELECT ON Course FROM scott;
```

User scott is not allowed to select from Course and user joan is not allowed to select from Course

References

Elmasri R., Navathe S. B., *Database Systems*, chapter 25.2

Ramakrishnan R., Gehrke J., *Database Management Systems*, chapters 21.1, 21.2, 21.3

https://sai.uow.edu.au/oradocs/
SQL Reference, GRANT statement

https://sai.uow.edu.au/oradocs/
SQL Reference, REVOKE statement