

9.10.1 දත්තපාදකයක දත්ත නිර්මාණය කිරීම සහ කළමනාකරණය සඳහා විශ්ලිත විමසුම් භාෂාව යොදාගැනීම (Using Structured Query Language – SQL to create and manage data in a database)

විශ්ලිත විමසුම් භාෂාව (Structured Query Language)



පුළුල් පරාසයක විභිදෙන දහසකුත් දත්ත සමූහයක් , යම්කිසි කාර්යයකට හැසිරවීම ලෙහෙසි පහසු දෙයක් නොවෙයි.මෙවැනි අවස්ථා වල පරිගණක දත්ත පාදක (Database) යොදා ගැනීමෙන් ඔබට ,දත්ත ගබඩා කිරීම, ආපසු ලබා ගැනීම සහ වෙනස් කිරීම වඩාත් පලදායී හා පහසු කාර්යයක් කර ගත හැක.

මුල් කාලයේ දත්ත පාදක (Database) සමග ගනුදෙනු කිරීමට ක්‍රමලේඛනය (programming) ගැන පුළුල් දැනුමක් අවශ්‍ය විය. ඒ මන්ද යත් මේ සඳහා භාවිතා කළ ක්‍රමලේඛන (programs) සංකීර්ණ වූ බැවින් දත්ත පාදක පරිපාලකයට ක්‍රමලේඛනය පිළිබඳ හසල දැනුමක් උවමන වීමයි .එබැවින් දත්ත පාදක (Database) භාවිතයට ක්‍රමලේඛශීලීන්ගේ (programmers) අනිවාර්ය මැදිහත්වීම අවශ්‍ය විය . දත්ත පාදකයට සම්බන්ධ සෑම අලුත් අවශ්‍යතාවයකදීම අලුත් පරිගණක වැඩසටහනක් නිර්මාණය කිරීමට සිදුවිය.

උදාහරණයක් වශයෙන්, විශ්වවිද්‍යාල දත්ත පාදකයට (database) සෑම වසරකම ඉහලම ලකුණු ලබාගත් ශිෂ්‍යයන්ගේ නම් සහ ලකුණු අකාරාදී පිළිවෙලට අවශ්‍ය වුවහොත්, එවැනි වැඩසටහනක් (program) නොමැතිනම් , ඒ සඳහා අලුත් වැඩසටහනක් නිර්මාණය කිරීමට සිදු විය. මෙය ඉතාම අකාර්යක්ෂම ක්‍රියාවලියක් වූ අතර දත්ත හසුරුවාලීමට තිබුණු අවශ්‍යතා ප්රමාණාත්මක ලෙස ඉහළ යාම හේතුවෙන් මේ ගැටළුව තවත් සංකීර්ණ විය. මේ වනවිට සිටි ක්‍රමලේඛ ශීලීන් ප්රමාණය එම අවශ්‍යතාවය සපුරන්නට තරම් ප්රමාණවත් නොවුණි. මේ නිසා ක්‍රමලේඛ දැනුමින් තොරව, සාමාන්‍ය පුද්ගලයෙකුට දත්ත පාදක භාවිත කරමින් තමන්ගේ අවශ්‍යතා ඉටුකරගත හැකි භාෂාවක අවශ්‍යතාවය වඩාත් ඉස්මතු විය.

බොහෝ අවස්ථා වල දත්ත පදකයක (Database) පවතින දත්ත සමුදායෙන් අපට එක් අවස්ථාවකදී අවශ්‍ය වන්නේ

කොටසක් පමණයි. ඒ දත්ත කොටස් ලබාගැනීම සඳහා අපි දත්ත පාදකය (database) වෙත යොමු කරන පැනය “විමසුමක්” (“query”) ලෙස හඳුන්වනු ලැබේ. ක්රමලේඛ ශීල්පීන් පරිගණක වැඩසටහන් ලෙස නිර්මාණයකරනු ලබන්නේ මෙම විමසුම් හෙවත් “query”ය. දත්ත ගබඩා හැසිරවීම සඳහා query පිළිබඳ දැනුවත් විය යුතුය. සංකීර්ණ දත්ත පාදක පහසුවෙන් හැසිරවීමට (query හැසිරවීම) සඳහා සරල භාෂාවක් නිර්මාණය කිරීම අත්‍යවශ්‍ය කාලීනව අවශ්‍යතාවක් විය.

පසු කාලයේදී දත්ත ගබඩා සඳහා සාමාන්‍ය පුද්ගලයන්ට භාවිත කරන්න පහසු සරල විමසුම් භාෂා (“query language”) විශාල ගණනක් නිර්මාණය විය. මේ අතරින් IBM ආයතනය විසින් 1970 දී හඳුන්වාදෙන ලද ව්‍යුහගත විමසුම් භාෂාව (“Structured Query Language – SQL”) අද දත්තපාදක කළමනාකරණය පද්ධති (“Database Management Systems”) වල බහුලව භාවිතා කෙරෙමින් පවතී. අද වනවිට SQL වල බහුලව භාවිත වෙන උපකොටස් තුනක් මේ ලිපිය තුළින් ආවරණය කෙරෙනු ඇත.

දත්ත අර්ථ නිරූපණ භාෂාව (Data Definition Language – DDL) Database එකක් තුළ දත්ත ගබඩා කිරීමට අවශ්‍ය සැකිල්ල විස්තර කිරීමට දත්ත අර්ථ නිරූපණ භාෂාව (DDL) භාවිතා කරනු ලැබේ. ඒ සඳහා බහුලව භාවිතා කරන විධාන කීපයක් විස්තරාත්මකව පහත දැක්වේ.

CREATE TABLE

```
4
5 CREATE TABLE table_name
6 (
7 column_name1 data_type(size),
8 column_name2 data_type(size),
9 column_name3 data_type(size),
10 ....
11 );
```

රූපයේ පරිදි මෙම විධානය මගින් දත්ත ගබඩා කිරීමට අවශ්‍ය සැකිල්ල නිර්මාණයකළ හැකිය. දත්ත සිරස් සහ තිරස් පෙළ “rows and columns” ලෙස සංවිධානය කරන්නේ මෙම සැකිල්ලෙනි., එය “table” එකක් ලෙස හඳුන්වනු ලැබේ . උදාහරණයක් මගින් මෙය වඩාත් පැහැදිලිව තේරුම් ගත හැකිය.

```
8 CREATE TABLE Persons
9 (
10 LastName varchar(255),
11 FirstName varchar(255),
12 Age int,
13 Address varchar(255),
14 City varchar(255),
15 );
```

මේ table එකේ නම “Persons”. එහි columns (තීරු) ආකාරයට සැලකුවොත් පුද්ගලයකුගේ “LastName, FirstName, Age, Address සහ City” යන දත්ත ගබඩා කළ හැකිය. columns සඳහා , ගබඩා කළ හැකි දත්ත වර්ගය “data types” ගබඩා කිරීමට ප්රථමයෙන් ලබාදිය යුතුයි . “int සහ varchar()” එවනි උදාහරණ දෙකක් පමණයි. “int” තුළ සංඛ්‍යාවක් සහ “varchar(255)” තුළ අක්ෂර 255 කට සීමාවූ ඕනෑම සංඛ්‍යා අකුරු මිශ්රනයක් ගබඩා කළ හැකිය.

ඉහත දත්ත අර්ථ නිරූපණ භාෂාව මගින් නිර්මාණය කළ , දත්ත කිසිවක් ගබඩා කර නැති Persons table එක පහත

ස්වරූපයෙන් දැක්වේ.

| LastName | FirstName | Age | Address | City |
|----------|-----------|-----|---------|------|
| | | | | |

```
7 CREATE TABLE Persons
8 (
9   PersonID int PRIMARY KEY,
10  LastName varchar(255),
11  FirstName varchar(255),
12  Age int,
13  Address varchar(255),
14  City varchar(255)
15  DepartmentID int FOREIGN KEY
16 );
```

ඉහත සකස් කළ Persons table එකට රූපයේ පරිදි තවත් විශේෂ columns දෙකක් එකතු කළ හැකිය. PRIMARY KEY විධානයෙන් ගබඩා කරන එක තීරුවක් තවත් තීරුවකින් (rows), වෙන්කොට හඳුනාගැනීමට PersonID column එක භාවිතාවේ. එමනිසා මෙම විධානයෙන් PersonID column එක කිසිවිටකත් හිස් විය නොහැකි බවත් සමාන අගයන් පැවතිය නොහැකි බවත් වක්රව සඳහන් කෙරෙයි. FOREIGN KEY විධානයෙන් Persons table එක තවත් එවැනි table එකකට සම්බන්ධ කෙරෙයි. DepartmentID column එක මෙම table දෙකටම පොදු වන අතර tables කිහිපයකින් දත්ත ලබාගැනීමට මෙය ඉතා ප්රයෝජනවත් වේ.

විශේෂ යෙදුම් කිහිපයක් (Some Column Constraints in SQL)

```
7 CREATE TABLE Persons
8 (
9   LastName varchar(255) NOT NULL,
10  FirstName varchar(255),
11  Age int CHECK (Age BETWEEN 0 AND 120),
12  Address varchar(255) DEFAULT 'Colombo',
13  City varchar(255)
14 );
```

- NOT NULL, CHECK සහ DEFAULT විධාන, columns තුළ ගබඩා කරන ලද දත්ත මත අමතර සීමා කීපයක් පැනවීමට යොදාගනු ලැබේ.
- දත්ත ඇති සෑම තීරුවකම , LastName පෙළ තුළ සැමවිටම අගයක් තිබිය යුතු බව දැක්වීමට NOT NULL විධානය භාවිතා කෙරෙයි.
- CHECK විධානය මගින් ගබඩා කරන දත්ත යම් කොන්දේසියකට යටත්දැයි පරීක්ෂා කරනු ලැබේ. මෙම උදාහරණයෙන් පුද්ගලයකුගේ වයස 0 – 120 සීමාව තුළ පවතිනම් පමණක් එම දත්තය ගබඩා කිරීමට මෙම විධානය යොදාගෙන ඇත.
- Address පෙළ තුළට, දත්ත ඇතුළත් කිරීමේදී (INSERT) ,විශේෂ අගයක් ලබාදී නොමැති නම් “Colombo” යන පොදු අගය ගබඩා කරන ලෙස දැක්වීමට DEFAULT විධානය භාවිතා කෙරෙයි.

ALTER TABLE මෙම විධානය මගින් පෙර නිර්මාණය කළ table එකකට අලුත් columns එකතු කිරීමටත්, පවතින columns අඩු කිරීමට හෝ වෙනස් කිරීමේ හැකියාව පවතී. එක් එක් අවස්ථාව සඳහා භාවිතා වන විධාන සහ උදාහරණ පහත

දැක්වේ.

```
8 ALTER TABLE table_name
9 ADD column_name datatype
10
11 ALTER TABLE Persons
12 ADD DateOfBirth date
13
14 ALTER TABLE table_name
15 DROP COLUMN column_name
16
17 ALTER TABLE Persons
18 DROP COLUMN City
19
20 ALTER TABLE table_name
21 MODIFY COLUMN column_name datatype
22
23 ALTER TABLE Persons
24 MODIFY COLUMN Address varchar(350)
```

DROP TABLE

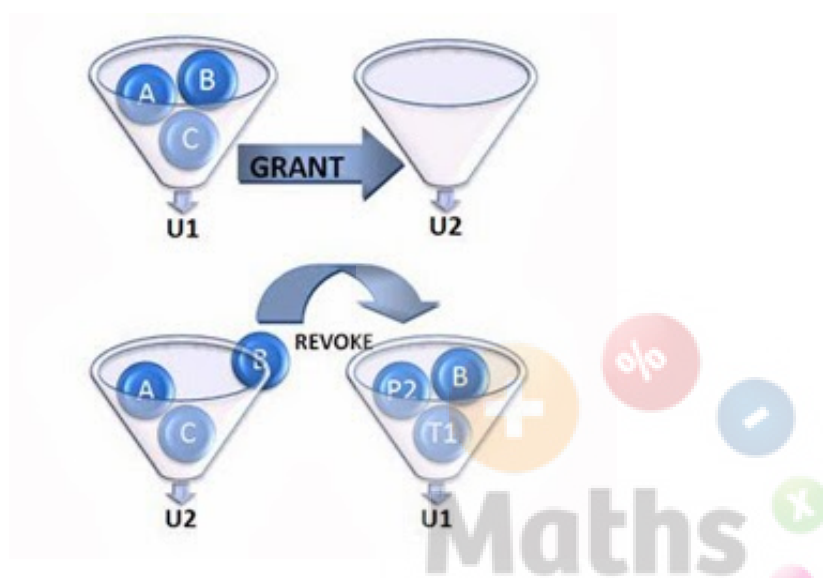
මෙම විධානය මගින් පෙර නිර්මාණය කළ tables සම්පූර්ණයෙන්ම මකා දැමිය හැකිය. ඒ සඳහා භාවිතා වන විධානය සහ උදාහරණයක් පහත දැක්වේ. විශේෂතාව වන්නේ table එකේ දත්ත හා සැකිල්ල සම්පූර්ණයෙන් දත්ත පදකයෙන් (database එකෙන්) මකා දැමීමයි.

```
7 DROP TABLE table_name
8
9 DROP TABLE Persons
```

SQL වල තවත් උපකොටස් දෙකක් ඊළඟ ලිපියෙන් සාකච්ඡා කරමු. The following two tabs change content below.

9.10.2 දත්තපාදකයක දත්ත නිර්මාණය කිරීම සහ කළමනාකරණය සඳහා ව්‍යුහගත විමසුම් භාෂාව යොදාගැනීම (Using Structured Query Language – SQL to create and manage data in a database)

දත්ත පාලන භාෂාව (Data Control Language – DCL)



දත්ත පාදකක් (Database) තුළ ගබඩා කරන ලද දත්ත, සියළුම පරිශීලකයන්ට සමාන වරප්රසාද සහිතව භාවිතා කිරීමට ඉඩ ලබා දීම එතරම් නුචණුකි ක්රියාවක් නොවෙයි. දත්ත පාලන භාෂාවේ විධාන (DCL commands) දත්ත පරිශීලනයට සීමා පැනවීමට යොදාගනී. මෙය දත්ත පාදකක් (Database) තුළ ඇති දත්ත වලට ආරක්ෂාව සැලසීමේ එක් ක්රියා මාර්ගයකි. මෙසේ අවසර ලබාදීමටත් අවලංගු කිරීමටත් හැකියාව ඇත්තේ දත්ත පාදක පරිපාලක (database administrator) හෝ දත්ත පාදක අයිතිකරුට (database owner) පමණයි. පහත දැක්වෙන විධාන දෙක ජර්මාන වශයෙන් යොදා ගනී.

- GRANT – දත්ත පාදක වස්තු (database objects) පරිශීලනයට වරප්රසාද ලබාදීම.
- REVOKE – ලබාදුන් වරපසාද නැවත අවලංගු කිරීම.

```
5 GRANT privilege_name TO user_name;
6
7 GRANT privilege_names
8 ON object_name
9 TO role_name;
10
11 GRANT role_name TO user_name;
12
13 REVOKE privilege_name FROM user_name;
14
15 REVOKE role_name FROM user_name;
```

මෙම විධාන දෙක භාවිතා කරන ආකාරය පහත දැක්වේ.

- **privilege_name** – පරිශීලකයන්ට ලබාදෙන වරප්‍රසාද වර්ගය (privilege type) උදාහරණ: SELECT, INSERT, DELETE
- **object_name** – වරප්‍රසාද (privilege) වලට වන දත්ත පාදක වස්තුව (database objects) උදාහරණ: වගු (table), ක්‍රියා පටිපාටි (procedure)
- **user_name** – වරප්‍රසාද ලබන්නාගේ නම
- **role_name** – වරප්‍රසාද කිහිපයක එකතුවක් එකවර පරිශීලකයකුට ලබාදීමට භාවිතා කරයි.

උදාහරණ මගින් මෙය වඩාත් පැහැදිලි කර ගනිමු. උදාහරණය සඳහා භාවිත කර තිබෙන්නේ Oracle දත්ත පාදකය වේ. දත්ත ආරක්ෂාව අතින් ඉතා ඉහල දත්ත පාදකයක් වන Oracle අධියන හා පර්යේෂණ කටයුතු සඳහා Express Edition කිසිදු ගෙවීමකින් තොරව භාවිත කළ හැකිය. බාගත කිරීම : <http://www.oracle.com/technetwork/products/express-edition/downloads/index.html>

Oracle Express Edition ස්ථාපනය පිළිබඳ ලිපියක් තුදුරු අනාගතයේදී පළවනු ඇත.

පළමුව දත්ත පාදක පරිපාලක ගිණුම (“SYS”) භාවිතයෙන් තවත් පරිශීලකයෙක් (user) නිර්මාණය කරගනිමු. ඒ සඳහා CREATE USER විධානය යොදා ගනී. ඔහු user_1 ලෙස හඳුන්වමු. user_1 හට දත්ත පාදකය පරිශීලනයට මුරපදයක් (password) ලබාදිය යුතුයි. මුරපදය test ලෙස සලකමු.

```
Enter user-name: SYS/ AS SYSDBA
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE USER user_1 IDENTIFIED BY test;
User created.
```

දැන් user_1 මගින් දත්ත පාදකය පරිශීලනයට (access) උත්සහ කරමු. එවිට, user_1 හට ඒ සඳහා ජර්මාණවත් වරප්‍රසාද නොමැති බව දත්ත පාදකය අපට දැනුම් දේ.

```
Enter user-name: user_1
Enter password:
ERROR:
ORA-01045: user USER_1 lacks CREATE SESSION privilege; logon denied
```

user_1 හට දත්ත පාදකය පරිශීලනයට ලබාදිය යුතු වරප්‍රසාදය CREATE SESSION නම් වේ. CREATE SESSION වරප්‍රසාදය user_1 වෙත ලබාදීම , නැවතත් දත්ත පාදක පරිපාලක ගිණුම (“SYS”) සිදුකළ යුතුයි. GRANT විධානය භාවිතයෙන් එය user_1 වෙත පවරනු ලැබේ.

```
Enter user-name: SYS/ AS SYSDBA
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> GRANT create session TO user_1;
Grant succeeded.
```

දැන් user_1 හට දත්ත පාදකයට ඇතුළු වීමට (login) පුළුවන්. user_1 හට “student” නමින් table එකක් නිර්මාණය කිරීමට අවශ්‍යයයි උපකල්පනය කරන්න. නමුත් user_1 හට “student” table එක නිර්මාණය කිරීමට තවමත් හැකියාවක් නැත . ඒ සඳහා තවත් වරප්‍රසාද කිහිපයක් ඔහුට අවශ්‍යයි.

```

Enter user-name: user_1
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE TABLE student(id INT, name VARCHAR(100),address VARCHAR(300));
CREATE TABLE student(id INT, name VARCHAR(100),address VARCHAR(300))
*
ERROR at line 1:
ORA-01031: insufficient privileges

```

ඒ සඳහා නැවතත් දත්ත පාදක පරිපාලක ගිණුමෙන් (“SYS”) දත්ත පාදකයට ඇතුළු විය යුතුයි. පහත පරිදි table නිර්මාණය කිරීමට වරප්‍රසාදය දෙකක් user_1 ට අවශ්‍ය වේ. Table එකක් නිර්මාණය කිරීමට “CREATE TABLE” වරප්‍රසාදය සහ ඒ සඳහා දත්ත පාදකය තුළ අවශ්‍යය මතකය (memory) ලබාදිය යුතුයි. දත්ත පාදක කළමනාකරණ පද්ධතිය(DBMS) ක්‍රියාත්මක වන විට එහි දත්ත පාදක පරිගණකයේ මතකය (memory) තුළ වෙනස්කම් (උදාහරණ : CREATE,INSERT,UPDATE,DELETE) සිදුකරන අතර නැවත දෘඪ තැටි (Hard Disk) වල ගබඩා කරයි. මෙහිදී “UNLIMITED TABLESPACE” මගින් user_1 ට සීමා රහිත මතකය ලබාදී තිබේ.

```

Enter user-name: SYS/ AS SYSDBA
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> GRANT create table TO user_1;

Grant succeeded.

SQL> GRANT unlimited tablespace TO user_1;

Grant succeeded.

```

දැන් user_1 හට id, name සහ address යන columns (කීරු) සහිත student table එක බාධාවක් නොමැතිව නිර්මාණය කර හැක.

```

Enter user-name: user_1
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE TABLE student(id INT, name VARCHAR(100),address VARCHAR(300));

Table created.

```

මෙම table එක user_1 හට අයත් නිසා SELECT, INSERT වැනි commands දත්ත පාදක පරිපාලක ගිණුමේ (“SYS”) මැදිහත්වීමකින් තොරව භාවිත කිරීමට පුළුවන. තවදුරත් වරප්‍රසාදය ලබාදීමට අවශ්‍යතාවක් නොමැත.

```

SQL> INSERT INTO student VALUES (1, 'a', 'colombo');

1 row created.

SQL> INSERT INTO student VALUES (2, 'b', 'galle');

1 row created.

```

```
SQL> SELECT * FROM student;
```

| ID | NAME | ADDRESS |
|----|------|---------|
| 1 | a | colombo |
| 2 | b | galle |

තවදුරටත් ගැඹුරින් අධ්‍යයනය සඳහා පහත උදාහරණය සලකා බලමු. ඒ සඳහා දත්ත පාදක පරිපාලක ගිණුමේ (“SYS”) යටතේ “university” සහ “lecturer” ලෙස තවත් table දෙකක් නිර්මාණය කර ගනිමු.

```
Enter user-name: SYS/ AS SYSDBA
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE TABLE university (id int, name varchar(100), location varchar(300));

Table created.

SQL> CREATE TABLE lecturer (id int, name varchar(100), department varchar(300));

Table created.
```

user_1 ලෙස දත්ත පාදකයට ඇතුළු වී (login) මෙම table දෙක පරිශීලනය කිරීමට උත්සහ ගතහොත්, එවැනි නම් සහිත table නොපවතින බව දත්ත පාදකය අපට දැනුම් දේ. මෙලෙස දත්ත පාදකය වැලැක්වීමට හේතුව, user_1, හට අයත් නොවන දත්ත පාදක පරිපාලක ගිණුමේ (“SYS”) සතු කාලපයේ tables දැකීමයි.

```
Enter user-name: user_1
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SELECT * FROM university;
SELECT * FROM university
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> SELECT * FROM lecturer;
SELECT * FROM lecturer
*
ERROR at line 1:
ORA-00942: table or view does not exist
```

user_1 හට මෙම දත්ත පාදක පරිශීලනය කිරීමට දත්ත පාදක පරිපාලක ගිණුමෙන් (“SYS”) ඒ සඳහා වරප්‍රසාද ලබාදිය යුතුයි. නමුත් SELECT, INSERT වැනි මෙහෙයුම් වරප්‍රසාද එකින් එක user ට ලබාදීම අකාර්යක්ෂමයි. වරප්‍රසාද කිහිපයක එකතුවක් එකවර පරිශීලකයකුට ලබාදීමට “role” භාවිතා කරයි.

පහත රූපයේ “CREATE ROLE” විධානය මගින් “user_role_1” සහ “user_role_2” නමැති role දෙකක් නිර්මාණය කර තියෙනවා. GRANT විධානය මගින් SELECT සහ INSERT මෙහෙයුම් වරප්‍රසාද එම user_role වලට ලබාදේ.

මෙහිදී lecturer සහ university යන වගු (tables) “SYS” user ට අයත් බව පෙන්වීමට ඒවා “SYS.lecturer” සහ “SYS.university” ලෙස විශේෂයෙන් හඳුන්වනු ලබනවා.

තවද lecturer නමැති table එකට SELECT සහ INSERT යන වරප්රසාද දෙකම ලබාදී ඇති නමුත් university table එකට ලබාදී ඇත්තේ SELECT වරප්රසාදය පමණයි.

```
Enter user-name: SYS/ AS SYSDBA
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> CREATE ROLE user_role_1;
Role created.

SQL> GRANT SELECT, INSERT ON SYS.lecturer TO user_role_1
2 ;

Grant succeeded.

SQL> CREATE ROLE user_role_2;
Role created.

SQL> GRANT SELECT ON SYS.university TO user_role_2;
Grant succeeded.
```

පහත පරිදි නිර්මාණය කරගත් roles දෙක user_1 හට ලබාදීමෙන් ඒවා තුල ඇති එක් එක් වරප්රසාද එකවර user_1 හට හිමිවේ.

```
SQL> GRANT user_role_1 TO user_1;
Grant succeeded.

SQL> GRANT user_role_2 TO user_1;
Grant succeeded.
```

උදාහරණය වඩාත් පැහැදිලි වීමට මෙම tables දෙකට ආදර්ශ දත්ත කිහිපයක් ඇතුලත් කරමු.

```
SQL> INSERT INTO university VALUES (1, 'moratuwa', 'katubada');
1 row created.

SQL> INSERT INTO lecturer VALUES (1, 'bbb', 'IT');
1 row created.
```

ලෙස දත්ත පාදකයට ඇතුළු වී (login) ,පහත විධාන වල ප්රතිපල නිරීක්ෂණය කරමු.

```

Enter user-name: user_1
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SELECT * FROM SYS.university;

      ID
-----
NAME
-----
LOCATION
-----

      1
moratuwa
katubada

```

```
SQL> SELECT * FROM SYS.lecturer;
```

```

      ID
-----
NAME
-----
DEPARTMENT
-----

      1
bbb
IT

```

table දෙකෙන්ම දත්තලකභා (SELECT) ගැනීමට user_1 ට වරප්රසාද ඇත.

```

SQL> INSERT INTO SYS.lecturer VALUES (2, 'ccc', 'IS');
1 row created.

SQL> INSERT INTO SYS.university VALUES (2, 'colombo', 'colombo');
INSERT INTO SYS.university VALUES (2, 'colombo', 'colombo')
*
ERROR at line 1:
ORA-01031: insufficient privileges

```

නමුත් user_1 හට දත්ත ඇතුලත් (INSERT) කළ හැක්කේ SYS.lecturer table එකට පමණි. ලබාදුන් වරප්රසාද අහෝසි කිරීමට REVOKE විධානය එක පහත පරිදි භාවිතා කිරීමට පළුවන. REVOKE මගින් වරප්රසාද එකින් එක අහෝසි කිරීමටත්, role එකක් අහෝසි කිරීම මගින් එකවර වරප්රසාද කිහිපයක් එකවර අහෝසි කිරීමට පුළුවන.

```

Enter user-name: SYS/ AS SYSDBA
Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> REVOKE user_role_1 FROM user_1;

Revoke succeeded.

SQL> REVOKE create table FROM user_1;

Revoke succeeded.

```

දෘෂ්‍ය අර්ථ භාෂාව (Visual Definition Language – VDL)

මෙමගින් user views සහ සංකල්පිත පරිපාටි “conceptual schema” අතරැති සම්බන්දතාව විස්තර කෙරෙනවා.

The following two tabs change content below.

