April 2021

SQL Basics: DDL

Introduction

What is SQL ?

What Can SQL do?

What is RDBMS?

SQL Command:

SQL defines following ways to manipulate data stored in an RDBMS.

- **DDL:** Data Definition Language
  - DDL commands changes the structure of the table like creation of table, altering table, deleting a table etc. All DDL commands are auto-committed. That means it saves all the changes permanently in the database.

Command	Description
create	to create new table or database
alter	for alteration
truncate	delete data from table
drop	to drop a table
rename	to rename a table

SQL Command:

- **DML:** Data Manipulation Language
  - DML commands are used for manipulating the data stored in the table and not the table itself. DML commands are not auto-committed. It means changes are not permanent to database, they can be rolled back.

Command	Description
insert	to insert a new row
update	to update existing row
delete	to delete a row
merge	merging two rows or two tables

- **DQL:** Data Query Language
  - It is used to fetch data from tables based on conditions that we can easily apply.

Creating and Managing Tables

- At the end of this session, you would be able to do the following:
  - Describe the main database objects
  - Create tables
  - Describe the data types that can be used when specifying column definition
  - Alter table definitions
  - Drop, rename, and truncate tables

Database Objects

Object	Description
Table	Basic unit of storage; composed of rows and columns
View	Logically represents subsets of data from one or more tables
Sequence	Numeric value generator
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

The CREATE TABLE Statement

- You must have:
  - CREATE TABLE privilege
  - A storage area

- You specify:
  - Table name
  - Column name, column data type, and column size

```
CREATE TABLE [schema.]table
             (column datatype [DEFAULT expr][, ...]);
```

Naming Rules

- Table names and column names:
  - Must begin with a letter
  - Must be 1–30 characters long
  - Must contain only A–Z, a–z, 0–9, _, $, and #
  - Must not duplicate the name of another object owned by the same user
  - Must not be an Oracle server reserved word

Creating Tables

- Create the table.

```
CREATE TABLE dept
            (deptno  NUMBER(2),
             dname   VARCHAR2(14),
             loc VARCHAR2(13));
Table created.
```

- Confirm table creation.

```
DESCRIBE dept
```

Name	Null?	Type
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

Tables in the Oracle Database

- User Tables:
  - Are a collection of tables created and maintained by the user
  - Contain user information

- Data Dictionary:
  - Is a collection of tables created and maintained by the Oracle Server
  - Contain database information

Querying the Data Dictionary

- See the names of tables owned by the user.

```
SELECT table_name
FROM   user_tables ;
```

- View distinct object types owned by the user.

```
SELECT DISTINCT object_type
FROM   user_objects ;
```

- View tables, views, synonyms, and sequences owned by the user.

```
SELECT *
FROM   user_catalog ;
```

Data Types

Data Type	Description
VARCHAR2(size)	Variable-length character data
CHAR(size)	Fixed-length character data
NUMBER(p,s)	Variable-length numeric data
DATE	Date and time values
LONG	Variable-length character data up to 2 gigabytes
CLOB	Character data up to 4 gigabytes
RAW and LONG RAW	Raw binary data
BLOB	Binary data up to 4 gigabytes
BFILE	Binary data stored in an external file; up to 4 gigabytes
ROWID	A 64 base number system representing the unique address of a row in its table.

Creating a Table by Using a Subquery Syntax

- Create a table and insert rows by combining the CREATE TABLE statement and the AS subquery option.

```
CREATE TABLE table
       [(column, column...)]
AS subquery;
```

- Match the number of specified columns to the number of subquery columns.
- Define columns with column names and default values.

Creating a Table by Using a Subquery

```
CREATE TABLE  dept80
  AS
    SELECT   employee_id, last_name,
             salary*12 ANNSAL,
             hire_date
    FROM     employees
    WHERE    department_id = 80;
Table created.
```

```
DESCRIBE dept80
```

Name	Null?	Type
EMPLOYEE_ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
ANNSAL		NUMBER
HIRE_DATE	NOT NULL	DATE

The ALTER TABLE Statement

- Use the ALTER TABLE statement to:
  - Add a new column
  - Modify an existing column
  - Define a default value for the new column
  - Drop a column

The ALTER TABLE Statement

- Use the ALTER TABLE statement to add, modify, or drop columns.

```
ALTER TABLE table
ADD         (column datatype [DEFAULT expr]
            [, column datatype]...);
```

```
ALTER TABLE table
MODIFY      (column datatype [DEFAULT expr]
            [, column datatype]...);
```

```
ALTER TABLE table
DROP       (column);
```

Adding a Column

New column

JOB_ID

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE
149	Zlotkey	126000	29-JAN-00
174	Abel	132000	11-MAY-96
176	Taylor	103200	24-MAR-98

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

Adding a Column

- You use the ADD clause to add columns.

```
ALTER TABLE dept80
ADD         (job_id VARCHAR2(9));
Table altered.
```

- The new column becomes the last column.

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

Modifying a Column

- You can change a column's data type, size, and default value.

```
ALTER TABLE   dept80
MODIFY     (last_name VARCHAR2(30));
Table altered.
```

- A change to the default value affects only subsequent insertions to the table

Dropping a Column

- Use the DROP COLUMN clause to drop columns you no longer need from the table.

```
ALTER TABLE  dept80
DROP COLUMN  job_id;
Table altered.
```

Dropping a Table

- All data and structure in the table is deleted.
- Any pending transactions are committed.
- All indexes are dropped.
- You cannot roll back the DROP TABLE statement.

```
DROP TABLE dept80;
Table dropped.
```

Changing the Name of an Object

- To change the name of a table, view, sequence, or synonym, you execute the RENAME statement.

```
RENAME dept TO detail_dept;
Table renamed.
```

- You must be the owner of the object.

Truncating a Table

- The TRUNCATE TABLE statement:
  - Removes all rows from a table
  - Releases the storage space used by that table

```
TRUNCATE TABLE detail_dept;
Table truncated.
```

- You cannot roll back row removal when using TRUNCATE.
- Alternatively, you can remove rows by using the DELETE statement.

Database Objects

Object	Description
Table	Basic unit of storage; composed of rows and columns
View	Logically represents subsets of data from one or more tables
Sequence	Numeric value generator
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

What is a View?

EMPLOYEES Table:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALA
100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	240
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	170
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	170
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	90
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-99	IT_PROG	42
124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-99	ST_MAN	58
141	Trenna	Rajs	TRAJS	650.121.8009	17-OCT-95	ST_CLERK	35
142	Curtis	Davies	CDAVIES	650.121.2994	29-JAN-97	ST_CLERK	31
143	Randall	Matos	RMATOS	650.121.2874	15-MAR-98	ST_CLERK	26
144	Peter	Vargas	PVARGAS	650.121.2004	09-JUL-98	ST_CLERK	25
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-00	SA_MAN	105
174	Ellen	Abel	EABEL	011.44.1644.429267	11-MAY-96	SA_REP	110
176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	24-MAR-98	SA_REP	86
178	Kimberely	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	70
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	44
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-96	MK_MAN	130
202	Pat	Fay	PFAY	603.123.6666	17-AUG-97	MK_REP	60
205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	120
206	William	Gietz	WGIETZ	515.123.8181	07-JUN-94	AC_ACCOUNT	83

EMPLOYEE_ID	LAST_NAME	SALARY
149	Zlotkey	10500
174	Abel	11000
176	Taylor	8600

20 rows selected.

Why Use Views?

01 To restrict data access

02 To make complex queries easy

03 To provide data independence

04 To present different views of the same data

Simple Views and Complex Views

Feature	Simple Views	Complex Views
Number of tables	One	One or more
Contain functions	No	Yes
Contain groups of data	No	Yes
DML operations, through a view	Yes	Not always

Creating a View

- You embed a subquery within the CREATE VIEW statement.
- The subquery can contain complex SELECT syntax.

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view
  [(alias[, alias]...)]
 AS subquery
```

Creating a View

- Create a view, EMPVU80, that contains details of employees in department 80.

```
CREATE VIEW   empvu80
 AS SELECT   employee_id, last_name, salary
    FROM     employees
    WHERE    department_id = 80;
View created.
```

- Describe the structure of the view by using the DESCRIBE command.

```
DESCRIBE empvu80
```

Creating a View by using column aliases

- Create a view by using column aliases in the subquery.
- Select the columns from this view by the given alias names.

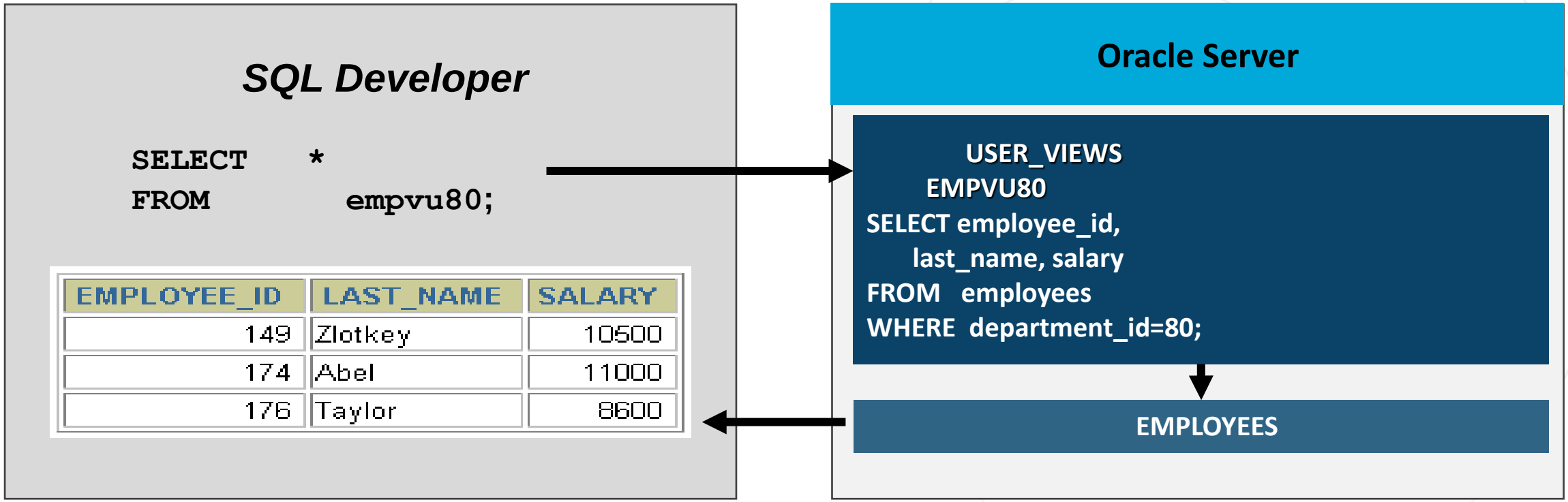
```
CREATE VIEW  salvu50
 AS SELECT  employee_id ID_NUMBER, last_name NAME,
            salary*12 ANN_SALARY
    FROM    employees
    WHERE   department_id = 50;
View created.
```

Retrieving Data from a View

```
SELECT *
FROM    salvu50;
```

ID_NUMBER	NAME	ANN_SALARY
124	Mourgos	69600
141	Rajs	42000
142	Davies	37200
143	Matos	31200
144	Vargas	30000

Querying a View



Modifying a View

- Modify the EMPVU80 view by using CREATE OR REPLACE VIEW clause. Add an alias for each column name.

```
CREATE OR REPLACE VIEW empvu80
  (id_number, name, sal, department_id)
AS SELECT  employee_id, first_name || ' ' || last_name,
           salary, department_id
   FROM    employees
   WHERE   department_id = 80;
View created.
```

- Column aliases in the CREATE VIEW clause are listed in the same order as the columns in the subquery.

Creating a Complex View

- Create a complex view that contains group functions to display values from two tables.

```
CREATE VIEW   dept_sum_vu
  (name, minsal, maxsal, avgsal)
AS SELECT   d.department_name, MIN(e.salary),
           MAX(e.salary),AVG(e.salary)
   FROM      employees e, departments d
   WHERE     e.department_id = d.department_id
   GROUP BY  d.department_name;
View created.
```

Removing a View

- You can remove a view without losing data because a view is based on underlying tables in the database.

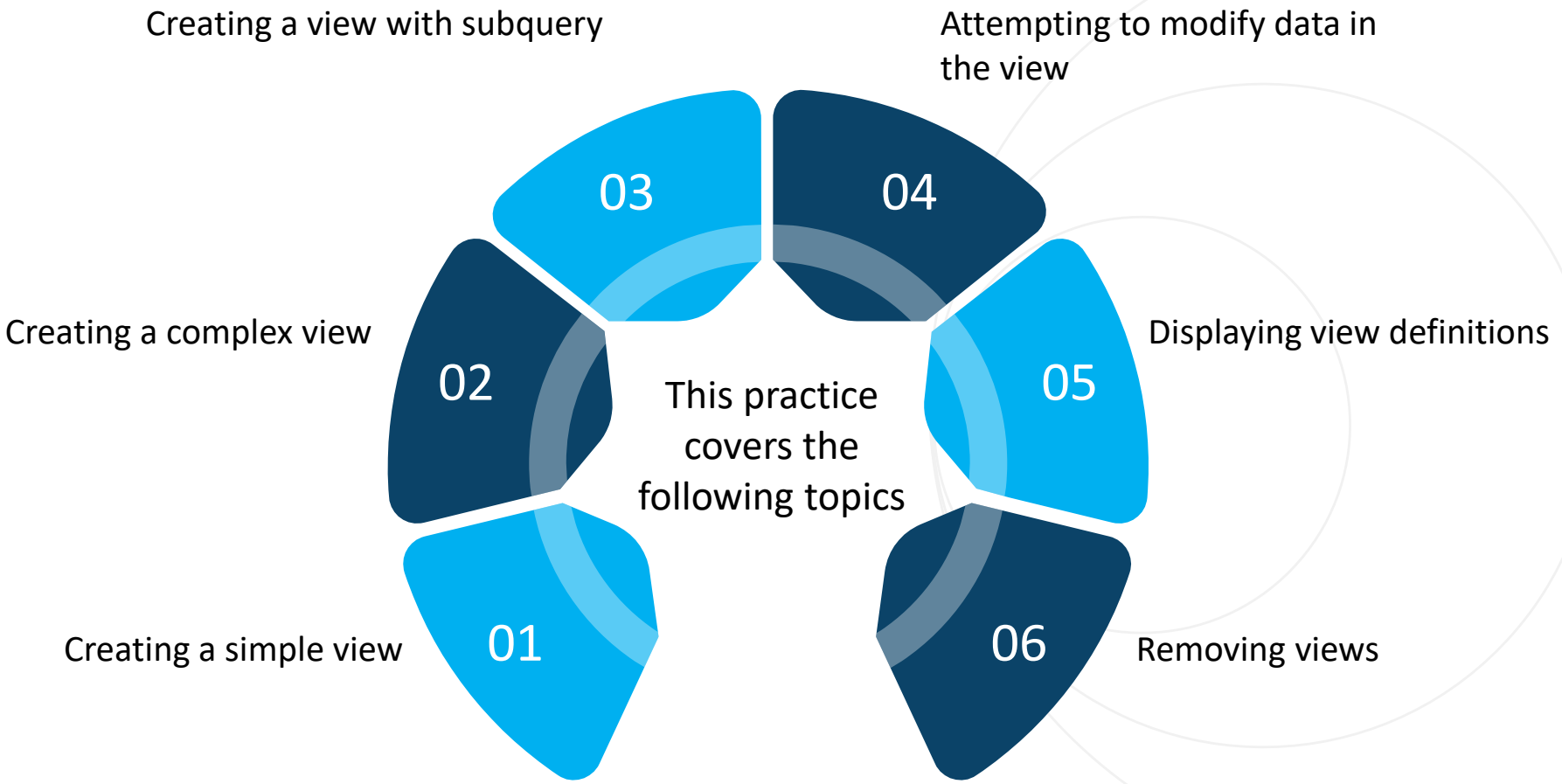
```
DROP VIEW view;
```

```
DROP VIEW empvu80;
View dropped.
```

Summary

- In this lesson, you should have learned that a view is derived from data in other tables or views and provides the following advantages:
  - Restricts database access
  - Simplifies queries
  - Provides data independence
  - Provides multiple views of the same data
  - Can be dropped without removing the underlying data

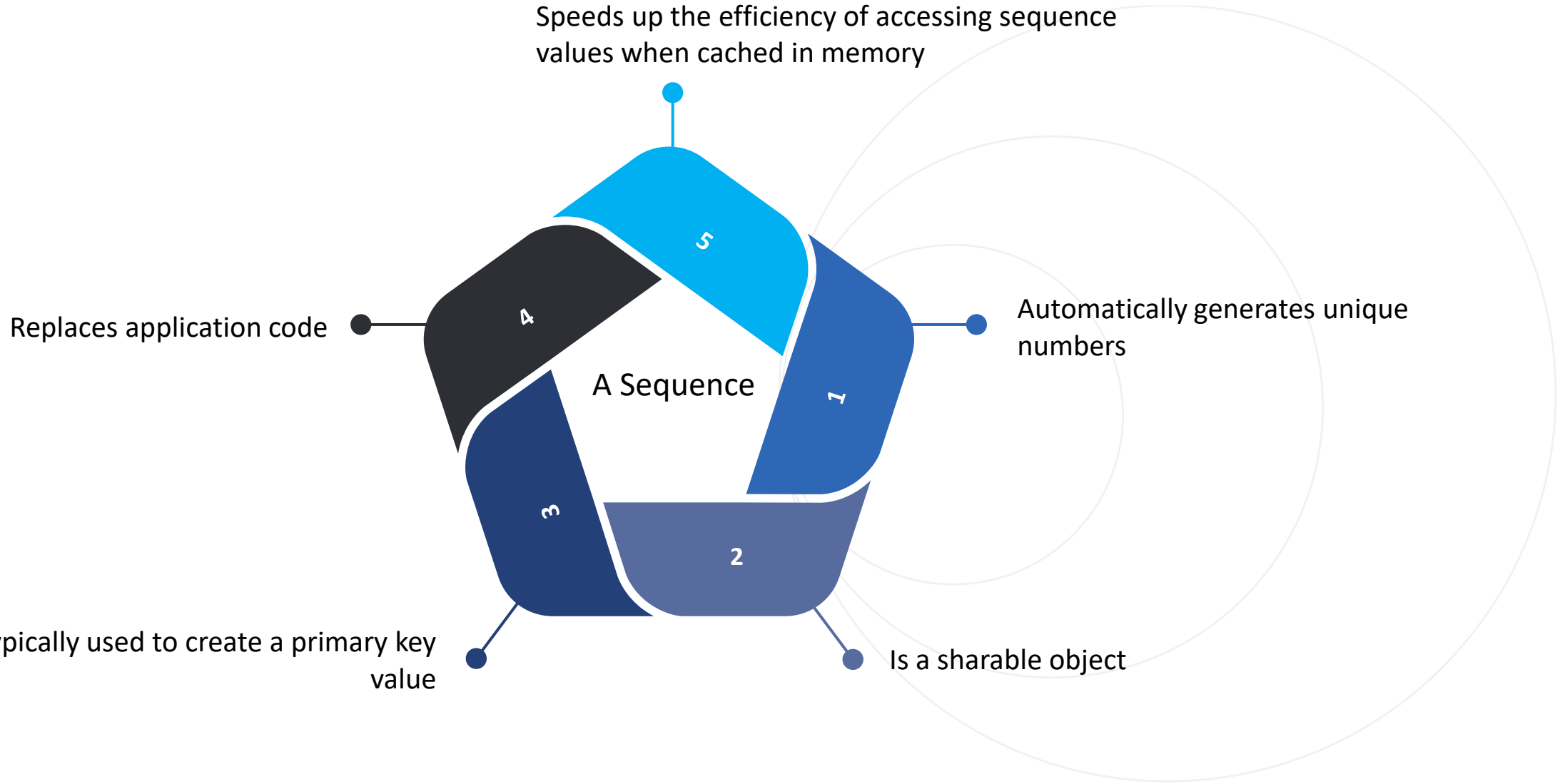
Practice Overview



Database Objects

Object	Description
Table	Basic unit of storage; composed of rows and columns
View	Logically represents subsets of data from one or more tables
Sequence	Numeric value generator
Index	Improves the performance of some queries
Synonym	Gives alternative names to objects

What Is a Sequence?



The CREATE SEQUENCE Statement Syntax

- Define a sequence to generate sequential numbers automatically:

```
CREATE SEQUENCE sequence
       [INCREMENT BY n]
       [START WITH n]
       [{MAXVALUE n | NOMAXVALUE}]
       [{MINVALUE n | NOMINVALUE}]
       [{CYCLE | NOCYCLE}]
       [{CACHE n | NOCACHE}];
```

Creating a Sequence

- Create a sequence named DEPT_DEPTID_SEQ to be used for the primary key of the DEPARTMENTS table.

```
CREATE SEQUENCE dept_deptid_seq
                INCREMENT BY 10
                START WITH 120
                MAXVALUE 9999
Sequence created.
```

Confirming Sequences

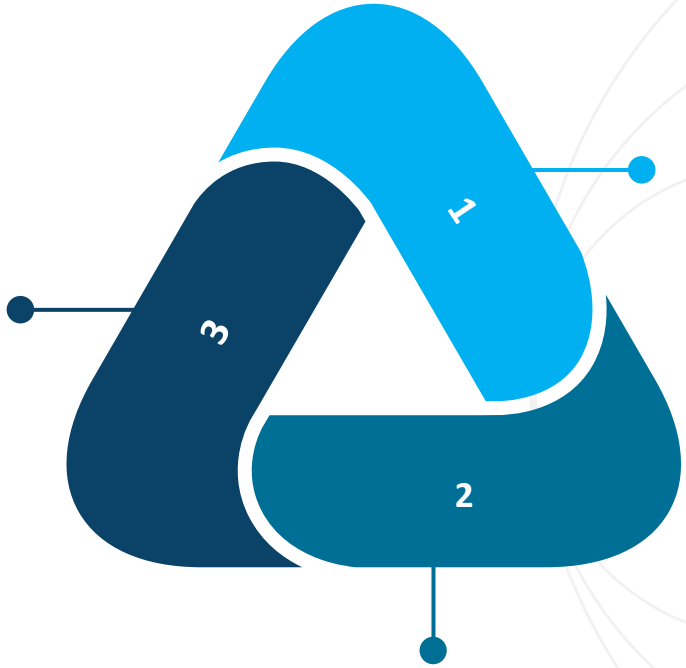
- Verify your sequence values in the USER_SEQUENCES data dictionary table.

```
SELECT   sequence_name, min_value, max_value,
         increment_by, last_number
FROM     user_sequences;
```

- The LAST_NUMBER column displays the next available sequence number if NOCACHE is specified.

NEXTVAL and CURRVAL Pseudocolumns

NEXTVAL must be issued for that sequence before CURRVAL contains a value.



NEXTVAL returns the next available sequence value. It returns a unique value every time it is referenced, even for different users.

CURRVAL obtains the current sequence value.

Using a Sequence

- Insert a new department named “Support” in location ID 2500.

```
INSERT INTO departments(department_id,
            department_name, location_id)
VALUES      (dept_deptid_seq.NEXTVAL,
            'Support', 2500);
1 row created.
```

- View the current value for the DEPT_DEPTID_SEQ sequence.

```
SELECT   dept_deptid_seq.CURRVAL
FROM     dual;
```

Modifying a Sequence

- Change the increment value, maximum value, minimum value, cycle option, or cache option.

```
ALTER SEQUENCE dept_deptid_seq
               INCREMENT BY 20
               MAXVALUE 999999
               NOCACHE
               NOCYCLE;
Sequence altered.
```

Removing a Sequence

- Remove a sequence from the data dictionary by using the DROP SEQUENCE statement.
- Once removed, the sequence can no longer be referenced.

```
DROP SEQUENCE dept_deptid_seq;
Sequence dropped.
```

infogain
Engineering Business Outcomes

Contact :

Training Team

Infogain India Pvt. Ltd.

Infogain Corporation, HQ
485 Alberto Way
Suite 100
Los Gatos, CA 95032
Phone: 408-355-6000

Irvine
41 Corporate Park
Suite 390
Irvine, CA 92606
Phone: 949-223-5100

Austin
111 W. Anderson Lane
Suite E336
Austin, TX 78752
Phone: 512-212-4070

Phoenix
21410 North 19th Ave
Suite 114
Phoenix, AZ 85027
Phone: 602-455-1860

London
Millbank Tower, Citibase 21-24
Millbank, office no 1.7
London SW1P 4DP
Phone: +44 (0)20 3355 7594

Noida
A-16 & 21, Sector 60
Gautam Budh Nagar
Noida – 201 301, India
Phone: +91 12 0244 5144

Mumbai
Unit No. 74, 2nd Floor, SDF 3
SEEPZ, Andheri East
Mumbai – 400 096, India
Phone: +91 22 6114 6969

Bengaluru
#7, 18th Main Road, 7th Block
Koramangala
Bengaluru - 560 095, India
Phone: +91 80 4110 4560

Dubai
Office No. 255, Building No. 17
Dubai Internet City
Dubai, UAE
Phone: +971-4-458-7336

Singapore
144 Robinson Road, #13-01
Robinson Square
Singapore 068908
Phone: +65 62741455