

Adventist University of Central Africa
Information Technology
PL/SQL Database Development
Group D

NAMES:

- UCYEYE Gabriella 27673
- TETA Juliet 27666
- UWIMPAYE Jeanette 27745

❖ INTRODUCTION:

This project demonstrates SQL concepts using a simple music database with three tables: Artists, Albums, and Songs. We use Oracle 10g to create tables, insert data, perform joins, create indexes, and define views, not forgetting to show the constraints.

❖ Database Design The database consists of three main tables:

- Artists: Stores artist information.
- Albums: Stores albums linked to artists.
- Songs: Stores songs linked to albums.
- Schema (simplified ERD): Artists (artist_id PK), Albums (album_id PK), Songs (song_id PK).

❖ CREATING TABLES

```
CREATE TABLE Artists (  
    artist_id INT PRIMARY KEY,  
    name VARCHAR(100) NOT NULL,  
    country VARCHAR(50) );
```

ORACLE Database Express Edition

User: SYSTEM

Home > Object Browser

Tables

ALBUMS

AQS_INTERNET_AGENTS

AQS_INTERNET_AGENT_PRIVS

AQS_QUEUES

AQS_QUEUE_TABLES

AQS_SCHEDULES

ARTISTS

CUSTOMER

DEF\$AQCALL

DEF\$AQERROR

DEF\$CALLDEST

DEF\$DEFAULTDEST

DEF\$DESTINATION

DEF\$ERROR

DEF\$LOB

DEF\$ORIGIN

DEF\$PROPAGATOR

DEF\$PUSHED_TRANSACTIONS

DEF\$TEMP\$LOB

HELP

LOGMINRC.DRNAME.UID.MAP

ARTISTS

Create

Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL

Add Column Modify Column Rename Column Drop Column Rename Copy Drop Truncate Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
ARTIST_ID	NUMBER	No	-	1
NAME	VARCHAR2(100)	No	-	-
COUNTRY	VARCHAR2(50)	Yes	-	-

1 - 3

Application Express 2.1.0.0.39

Copyright © 1999, 2006, Oracle. All rights reserved.

CREATE TABLE Albums (

album_id INT PRIMARY KEY,

title VARCHAR(100) NOT NULL,

release_year INT, artist_id INT,

FOREIGN KEY (artist_id) REFERENCES Artists(artist_id));

ORACLE Database Express Edition

User: SYSTEM

Home > Object Browser

Tables

ALBUMS

AQS_INTERNET_AGENTS

AQS_INTERNET_AGENT_PRIVS

AQS_QUEUES

AQS_QUEUE_TABLES

AQS_SCHEDULES

ARTISTS

CUSTOMER

DEF\$AQCALL

DEF\$AQERROR

DEF\$CALLDEST

DEF\$DEFAULTDEST

DEF\$DESTINATION

DEF\$ERROR

DEF\$LOB

DEF\$ORIGIN

DEF\$PROPAGATOR

DEF\$PUSHED_TRANSACTIONS

DEF\$TEMP\$LOB

HELP

LOGMINRC.DRNAME.UID.MAP

ALBUMS

Create

Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL

Add Column Modify Column Rename Column Drop Column Rename Copy Drop Truncate Create Lookup Table

Column Name	Data Type	Nullable	Default	Primary Key
ALBUM_ID	NUMBER	No	-	1
TITLE	VARCHAR2(100)	No	-	-
RELEASE_YEAR	NUMBER	Yes	-	-
ARTIST_ID	NUMBER	Yes	-	-

1 - 4

Application Express 2.1.0.0.39

Copyright © 1999, 2006, Oracle. All rights reserved.

CREATE TABLE Songs (

song_id INT PRIMARY KEY,

title VARCHAR(100) NOT NULL,
 duration INT,
 album_id INT,
 FOREIGN KEY (album_id) REFERENCES Albums(album_id));

The screenshot shows the Oracle Database Express Edition interface. On the left is a tree view of database objects, with 'SONGS' selected under the 'REPCAT\$' schema. The main area displays the 'SONGS' table structure with the following columns:

Column Name	Data Type	Nullable	Default	Primary Key
SONG_ID	NUMBER	No	-	1
TITLE	VARCHAR2(100)	No	-	-
DURATION	NUMBER	Yes	-	-
ALBUM_ID	NUMBER	Yes	-	-

At the bottom right of the table structure, it indicates '1 - 4'.

❖ INSERTING DATA

INSERT ALL

INTO Artists (artist_id, name, country) VALUES (1, 'Ed Sheeran', 'UK')

INTO Artists (artist_id, name, country) VALUES (2, 'Adele', 'UK')

INTO Artists (artist_id, name, country) VALUES (3, 'Drake', 'Canada')

SELECT * FROM dual;

ORACLE Database Express Edition

User: SYSTEM

Home > Object Browser

Tables

ARTISTS

Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL

Query Count Rows Insert Row

EDIT	ARTIST_ID	NAME	COUNTRY
	1	Ed Sheeran	UK
	2	Adele	UK
	3	Drake	Canada
row(s) 1 - 3 of 3			

Download

Language: en

Application Express 2.1.0.0.39
Copyright © 1990, 2006, Oracle. All rights reserved.

INSERT ALL

INTO Albums (album_id, title, release_year, artist_id) VALUES (1, 'Divide', 2017, 1)

INTO Albums (album_id, title, release_year, artist_id) VALUES (2, '25', 2015, 2)

INTO Albums (album_id, title, release_year, artist_id) VALUES (3, 'Scorpion', 2018, 3)

INTO Albums (album_id, title, release_year, artist_id) VALUES (4, 'Unreleased Album', 2020, 2) SELECT * FROM dual;

ORACLE Database Express Edition

User: SYSTEM

Home > Object Browser

Tables

ALBUMS

Table Data Indexes Model Constraints Grants Statistics UI Defaults Triggers Dependencies SQL

Query Count Rows Insert Row

EDIT	ALBUM_ID	TITLE	RELEASE_YEAR	ARTIST_ID
	1	Divide	2017	1
	2	25	2015	2
	3	Scorpion	2018	3
	4	Unreleased Album	2020	2
row(s) 1 - 4 of 4				

Download

Microsoft Edge

Application Express 2.1.0.0.39

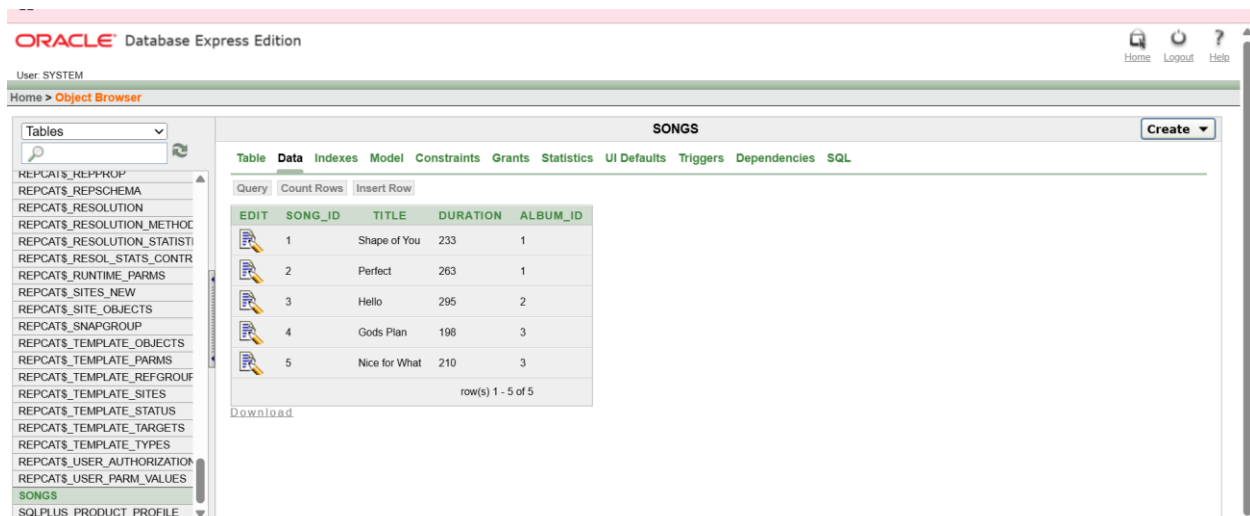
INSERT ALL

INTO Songs (song_id, title, duration, album_id) VALUES (1, 'Shape of You', 233, 1)

```

INTO Songs (song_id, title, duration, album_id) VALUES (2, 'Perfect', 263, 1)
INTO Songs (song_id, title, duration, album_id) VALUES (3, 'Hello', 295, 2)
INTO Songs (song_id, title, duration, album_id) VALUES (4, 'Gods Plan', 198, 3)
INTO Songs (song_id, title, duration, album_id) VALUES (5, 'Nice for What', 210, 3)
SELECT * FROM dual;

```



The screenshot shows the Oracle Database Express Edition interface. On the left is the 'Object Browser' pane with a tree view of database objects. The 'SONGS' table is selected. The main pane displays the 'SONGS' table with a 'Data' tab active. The table contains 5 rows of data. The columns are SONG_ID, TITLE, DURATION, and ALBUM_ID. The data is as follows:

EDIT	SONG_ID	TITLE	DURATION	ALBUM_ID
	1	Shape of You	233	1
	2	Perfect	263	1
	3	Hello	295	2
	4	Gods Plan	198	3
	5	Nice for What	210	3

At the bottom of the table, it says 'row(s) 1 - 5 of 5'. There is a 'Download' link below the table.

❖ QUERIES AND RESULTS

❖ INNER JOIN (Songs with their Albums and Artists)

```

SELECT s.title AS song, a.title AS album, ar.name AS artist FROM Songs s
INNER JOIN Albums a ON s.album_id = a.album_id
INNER JOIN Artists ar ON a.artist_id = ar.artist_id;

```

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save Run

```
SELECT s.title AS song, a.title AS album, ar.name AS artist
FROM Songs s
INNER JOIN Albums a ON s.album_id = a.album_id
INNER JOIN Artists ar ON a.artist_id = ar.artist_id;
```

Results Explain Describe Saved SQL History

SONG	ALBUM	ARTIST
Shape of You	Divide	Ed Sheeran
Perfect	Divide	Ed Sheeran
Hello	25	Adele
Gods Plan	Scorpion	Drake
Nice for What	Scorpion	Drake

5 rows returned in 0.01 seconds CSV Export

Language: en Application Exp Show desktop Copyright © 1990, 2006, Oracle All

❖ LEFT JOIN (All albums, even if no songs yet)

```
SELECT a.title AS album, s.title AS song
FROM Albums a
LEFT JOIN Songs s ON a.album_id = s.album_id;
```

ORACLE Database Express Edition

User: SYSTEM

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save Run

```
SELECT a.title AS album, s.title AS song
FROM Albums a
LEFT JOIN Songs s ON a.album_id = s.album_id;
```

Results Explain Describe Saved SQL History

ALBUM	SONG
Divide	Shape of You
Divide	Perfect
25	Hello
Scorpion	Gods Plan
Scorpion	Nice for What
Unreleased Album	-

6 rows returned in 0.01 seconds CSV Export

Application Exp Show desktop Copyright © 1990, 2006, Oracle All

❖ RIGHT JOIN

```
SELECT a.title AS album, s.title AS song
FROM Albums a
```

RIGHT JOIN Songs s ON a.album_id = s.album_id;

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the following query:

```
SELECT a.title AS album, s.title AS song
FROM Albums a
RIGHT JOIN Songs s ON a.album_id = s.album_id;
```

The results are displayed in a table with 5 rows:

ALBUM	SONG
Divide	Perfect
Divide	Shape of You
25	Hello
Scorpion	Nice for What
Scorpion	Gods Plan

5 rows returned in 0.00 seconds

❖ FULL JOIN

SELECT a.title AS album, s.title AS song

FROM Albums a

FULL OUTER JOIN Songs s ON a.album_id = s.album_id;

The screenshot shows the Oracle Database Express Edition interface. The SQL command window contains the following query:

```
SELECT a.title AS album, s.title AS song
FROM Albums a
FULL OUTER JOIN Songs s ON a.album_id = s.album_id;
```

The results are displayed in a table with 6 rows:

ALBUM	SONG
Divide	Shape of You
Divide	Perfect
25	Hello
Scorpion	Gods Plan
Scorpion	Nice for What
Unreleased Album	-

6 rows returned in 0.00 seconds

❖ Creating an Index

CREATE INDEX idx_album_title ON Albums(title);

❖ Creating a View

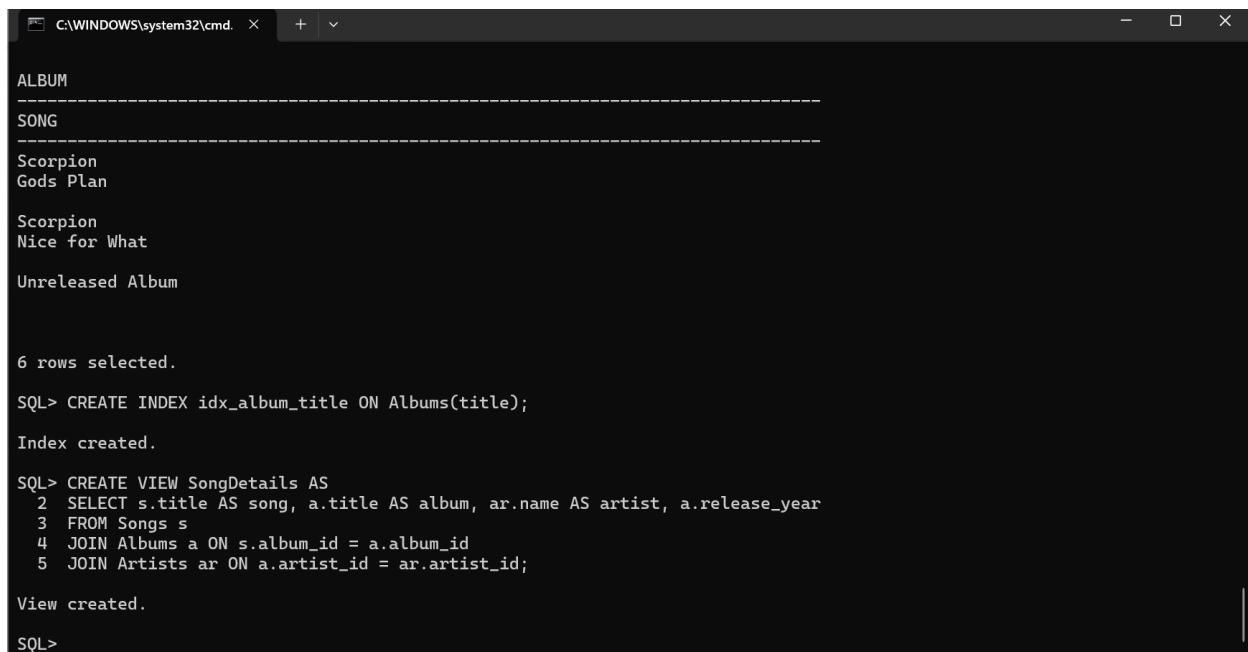
```
CREATE VIEW SongDetails AS
```

```
SELECT s.title AS song, a.title AS album, ar.name AS artist, a.release_year
```

```
FROM Songs s
```

```
JOIN Albums a ON s.album_id = a.album_id
```

```
JOIN Artists ar ON a.artist_id = ar.artist_id;
```



```
C:\WINDOWS\system32\cmd. x + v
ALBUM
-----
SONG
-----
Scorpion
Gods Plan
Scorpion
Nice for What
Unreleased Album

6 rows selected.

SQL> CREATE INDEX idx_album_title ON Albums(title);

Index created.

SQL> CREATE VIEW SongDetails AS
  2 SELECT s.title AS song, a.title AS album, ar.name AS artist, a.release_year
  3 FROM Songs s
  4 JOIN Albums a ON s.album_id = a.album_id
  5 JOIN Artists ar ON a.artist_id = ar.artist_id;

View created.

SQL>
```

❖ CONCLUSION

In conclusion, the artists, albums, and songs databases in Oracle 10g show how music data can be organized clearly and efficiently. Each artist has unique information, albums are linked to artists, and songs are linked to both albums and artists. Using SQL commands like INSERT, UPDATE, and SELECT, users can easily add, modify, or retrieve data while keeping everything accurate and organized. Indexes improve performance by speeding up data searches, while views provide a simplified way to display or access specific data without altering the original tables. Overall, it demonstrates how Oracle 10g helps manage complex information in a simple, reliable, and practical way.