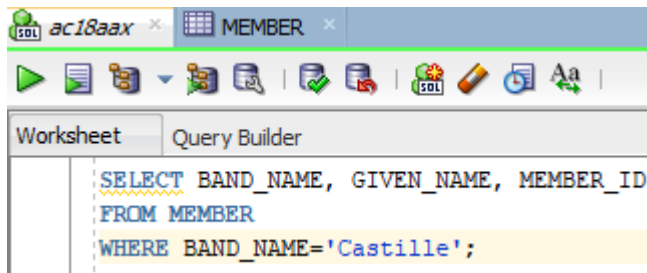


1. Report the Names of the members of the band 'Castille'.

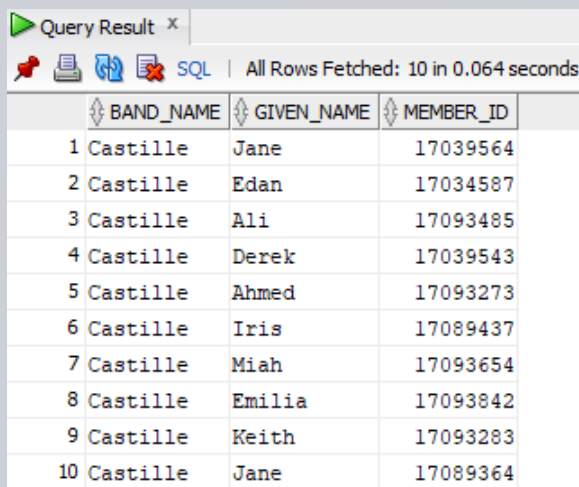
Screenshot of query



The screenshot shows a SQL query editor window with the following SQL code:

```
SELECT BAND_NAME, GIVEN_NAME, MEMBER_ID  
FROM MEMBER  
WHERE BAND_NAME='Castille';
```

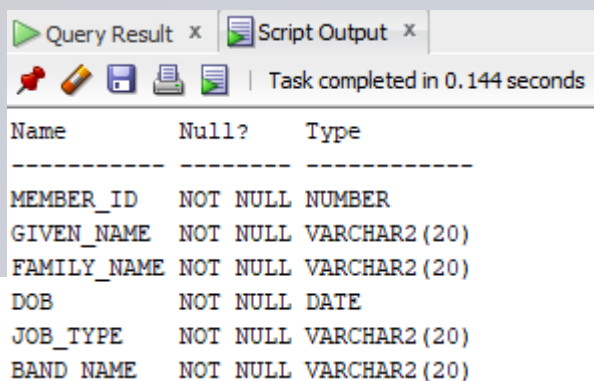
Screenshot of output:



The screenshot shows the results of the SQL query, displaying 10 rows of data. The columns are BAND_NAME, GIVEN_NAME, and MEMBER_ID. The data is as follows:

	BAND_NAME	GIVEN_NAME	MEMBER_ID
1	Castille	Jane	17039564
2	Castille	Edan	17034587
3	Castille	Ali	17093485
4	Castille	Derek	17039543
5	Castille	Ahmed	17093273
6	Castille	Iris	17089437
7	Castille	Miah	17093654
8	Castille	Emilia	17093842
9	Castille	Keith	17093283
10	Castille	Jane	17089364

Table description:

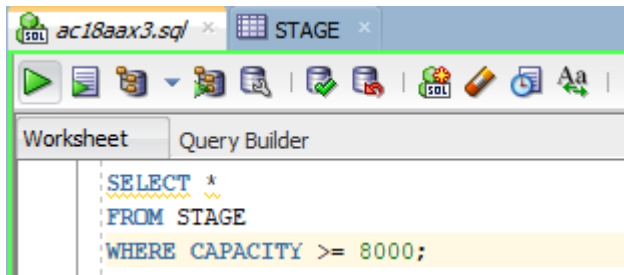


The screenshot shows the table description for the MEMBER table, displaying the column names, nullability, and data types. The data is as follows:

Name	Null?	Type
MEMBER_ID	NOT NULL	NUMBER
GIVEN_NAME	NOT NULL	VARCHAR2 (20)
FAMILY_NAME	NOT NULL	VARCHAR2 (20)
DOB	NOT NULL	DATE
JOB_TYPE	NOT NULL	VARCHAR2 (20)
BAND_NAME	NOT NULL	VARCHAR2 (20)

2. List the stage number and name of those stages that have a capacity of 8000 or more

Screenshot of query



Screenshot of output:

The screenshot shows the 'Query Result' window displaying the output of the query. The table has three columns: STAGE_NO, STAGE_NAME, and CAPACITY. The data rows are:

STAGE_NO	STAGE_NAME	CAPACITY
1	1 Pyramid	12000
2	2 Sonic	8000
3	5 Spartan	14000

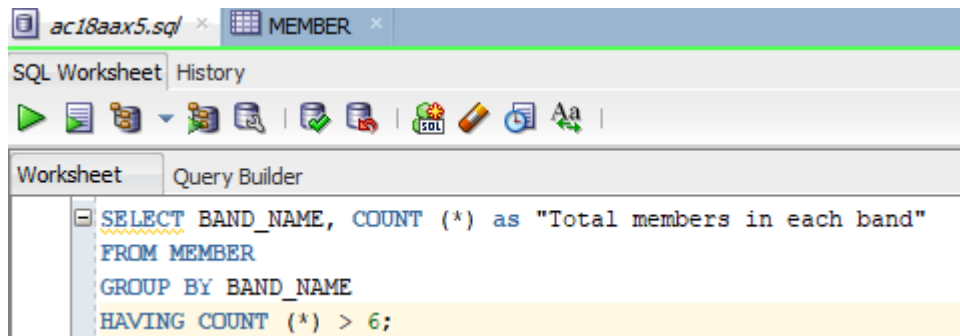
Table Description:

The screenshot shows the 'Script Output' window displaying the table description for the STAGE table. The output is as follows:

Name	Null?	Type
STAGE_NO	NOT NULL	NUMBER
STAGE_NAME		VARCHAR2 (20)
CAPACITY		NUMBER

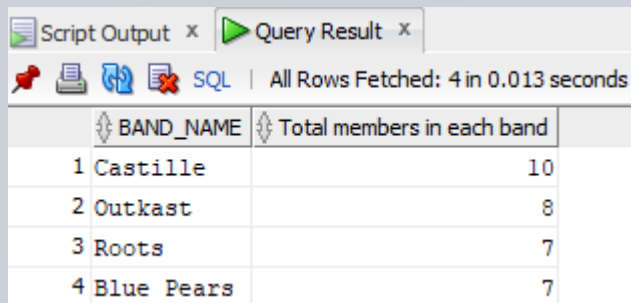
- List all band names and number of members in each band, for which the total members of the band exceeds 6.

Screenshot of query:



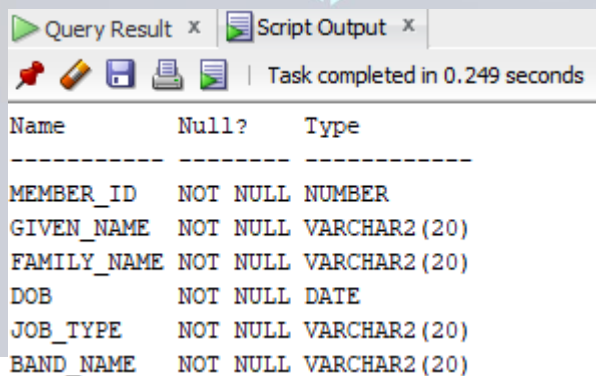
```
SELECT BAND_NAME, COUNT (*) as "Total members in each band"
FROM MEMBER
GROUP BY BAND_NAME
HAVING COUNT (*) > 6;
```

Screenshot of output:



BAND_NAME	Total members in each band
1 Castille	10
2 Outkast	8
3 Roots	7
4 Blue Pears	7

Table Description:



Name	Null?	Type
MEMBER_ID	NOT NULL	NUMBER
GIVEN_NAME	NOT NULL	VARCHAR2 (20)
FAMILY_NAME	NOT NULL	VARCHAR2 (20)
DOB	NOT NULL	DATE
JOB_TYPE	NOT NULL	VARCHAR2 (20)
BAND_NAME	NOT NULL	VARCHAR2 (20)



- List each stage by name, and the total number of members requiring access to each stage over the festival.

Screenshot of query:

```

SELECT STAGE.STAGE_NAME, COUNT (MEMBER.MEMBER_ID) AS "Total member(s) requiring access"
FROM STAGE
JOIN PERFORMANCE
JOIN MEMBER
ON MEMBER.BAND_NAME = PERFORMANCE.BAND_NAME
ON STAGE.STAGE_NO = PERFORMANCE.STAGE_NO
GROUP BY STAGE.STAGE_NAME;

```

Screenshot of output:

STAGE_NAME	Total member(s) requiring access
1 Sonic	10
2 StoneBridge	7
3 Spartan	7
4 Glade	6
5 Pyramid	14

Table Description:

```

Name      Null?     Type
-----
MEMBER_ID NOT NULL  NUMBER
GIVEN_NAME NOT NULL  VARCHAR2 (20)
FAMILY_NAME NOT NULL  VARCHAR2 (20)
DOB        NOT NULL  DATE
JOB_TYPE   NOT NULL  VARCHAR2 (20)
BAND_NAME  NOT NULL  VARCHAR2 (20)

Name      Null?     Type
-----
PERFORMANCE_ID NOT NULL  NUMBER
STAGE_NO      NUMBER
BAND_NAME     VARCHAR2 (20)
DATE_TIME     TIMESTAMP (6)

Name      Null?     Type
-----
STAGE_NO    NOT NULL  NUMBER
STAGE_NAME  VARCHAR2 (20)
CAPACITY    NUMBER

```

- List all the members of bands appearing on the stage called 'Spartan'. Join the family name and given name into a single column in the output.

Screenshot of query:

```

SELECT MEMBER.FAMILY_NAME || ', ' || MEMBER.GIVEN_NAME AS "Full Name"
FROM MEMBER, PERFORMANCE, STAGE
WHERE PERFORMANCE.BAND_NAME = MEMBER.BAND_NAME
AND STAGE.STAGE_NO = PERFORMANCE.STAGE_NO
AND STAGE.STAGE_NAME = 'Spartan';

```

Screenshot of output:

	Full Name
1	Stevens, Cole
2	Ross, Enzo
3	Choi, Mason
4	Pace, Alisson
5	Parsons, Marco
6	Wall, Alec
7	Leonard, Miya

Table Description:

```

Name      Null?     Type
-----
MEMBER_ID NOT NULL  NUMBER
GIVEN_NAME NOT NULL  VARCHAR2(20)
FAMILY_NAME NOT NULL  VARCHAR2(20)
DOB        NOT NULL  DATE
JOB_TYPE   NOT NULL  VARCHAR2(20)
BAND_NAME  NOT NULL  VARCHAR2(20)
Name      Null?     Type
-----
PERFORMANCE_ID NOT NULL  NUMBER
STAGE_NO       NUMBER
BAND_NAME      VARCHAR2(20)
DATE_TIME      TIMESTAMP(6)
Name      Null?     Type
-----
STAGE_NO  NOT NULL  NUMBER
STAGE_NAME VARCHAR2(20)
CAPACITY  NUMBER

```

- List the stage no, stage name and capacity and start date and time of all performances by the band 'Blue Pears.'

Screenshot of query:

```
SQL Worksheet History
[Icons]
Worksheet Query Builder
SELECT STAGE.STAGE_NO AS "STAGE NUMBER", STAGE.STAGE_NAME AS "STAGE NAME", STAGE.CAPACITY, TO_CHAR ( PERFORMANCE.DATE_TIME, 'DD-MM-YYYY HH:MI:SS') AS "START DATE AND TIME"
FROM STAGE
JOIN PERFORMANCE
ON STAGE.STAGE_NO = PERFORMANCE.STAGE_NO
WHERE PERFORMANCE.BAND_NAME = 'Blue Pears';
```

Screenshot of output:

Query Result x

SQL | All Rows Fetched: 1 in 0.016 seconds

	STAGE NUMBER	STAGE NAME	CAPACITY	START DATE AND TIME
1	5	Spartan	14000	13-01-2020 08:30:00

Table Description:

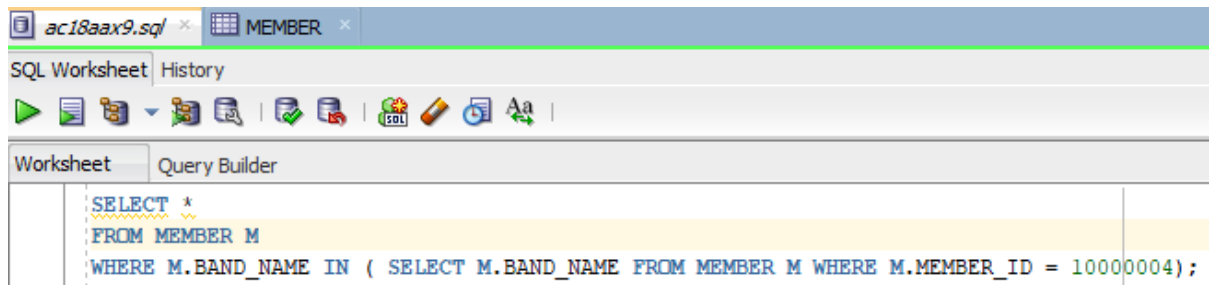
Script Output x

Task completed in 0.084 seconds

Name	Null?	Type
STAGE_NO	NOT NULL	NUMBER
STAGE_NAME		VARCHAR2 (20)
CAPACITY		NUMBER
PERFORMANCE_ID	NOT NULL	NUMBER
STAGE_NO		NUMBER
BAND_NAME		VARCHAR2 (20)
DATE_TIME		TIMESTAMP (6)

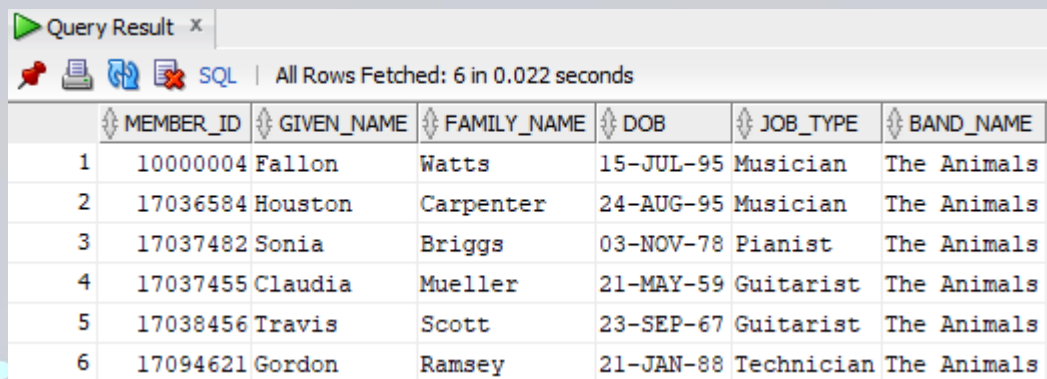
- Using a nested select list other member details of members who work in the same band as member number '10000004'.

Screenshot of query:



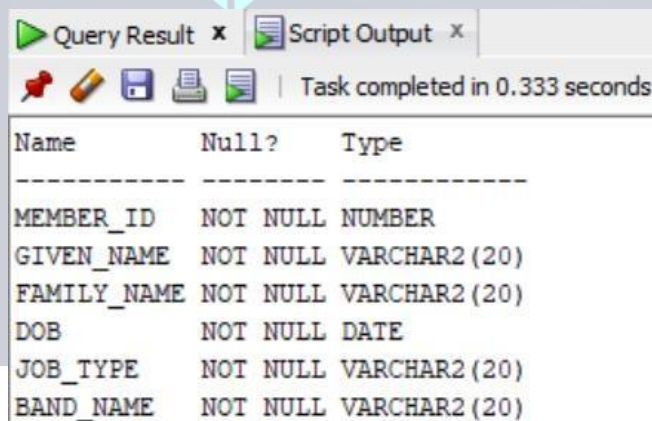
```
SELECT *
FROM MEMBER M
WHERE M.BAND_NAME IN ( SELECT M.BAND_NAME FROM MEMBER M WHERE M.MEMBER_ID = 10000004);
```

Screenshot of output:



	MEMBER_ID	GIVEN_NAME	FAMILY_NAME	DOB	JOB_TYPE	BAND_NAME
1	10000004	Fallon	Watts	15-JUL-95	Musician	The Animals
2	17036584	Houston	Carpenter	24-AUG-95	Musician	The Animals
3	17037482	Sonia	Briggs	03-NOV-78	Pianist	The Animals
4	17037455	Claudia	Mueller	21-MAY-59	Guitarist	The Animals
5	17038456	Travis	Scott	23-SEP-67	Guitarist	The Animals
6	17094621	Gordon	Ramsey	21-JAN-88	Technician	The Animals

Table Description:



Name	Null?	Type
MEMBER_ID	NOT NULL	NUMBER
GIVEN_NAME	NOT NULL	VARCHAR2 (20)
FAMILY_NAME	NOT NULL	VARCHAR2 (20)
DOB	NOT NULL	DATE
JOB_TYPE	NOT NULL	VARCHAR2 (20)
BAND_NAME	NOT NULL	VARCHAR2 (20)

- List the Name and Year of Birth of all members in alphabetic order by family name and given name, that have Claire Lascayne as their agent

Screenshot of query:

```

SELECT MEMBER.FAMILY_NAME || ',' || MEMBER.GIVEN_NAME AS "FULL NAME", TO_CHAR (MEMBER.DOB, 'DD-MM-YYYY') AS "AGE"
FROM MEMBER
JOIN BAND ON MEMBER.BAND_NAME = BAND.BAND_NAME
WHERE BAND.AGENT_NO = 101
ORDER BY MEMBER.FAMILY_NAME || ',' || MEMBER.GIVEN_NAME ASC;

```

Screenshot of output:

	FULL NAME	AGE
1	Barnes, Skylar	18-11-1988
2	Bates, Vanessa	03-02-1997
3	Briggs, Luke	13-12-1968
4	Carter, Frank	05-05-1966
5	Cox, Joe	24-09-1997
6	Escobar, Mayra	05-11-1992
7	Gates, Emmanuel	02-02-1992
8	Graves, Haylie	23-05-1992
9	Herring, Devonte	12-08-1992
10	Meza, Joseph	15-04-1988
11	Moses, Derek	16-06-1984
12	Rojas, Sage	15-11-1956
13	Rossi, Natalie	08-04-1994

thonyc.co.uk

Table Description:

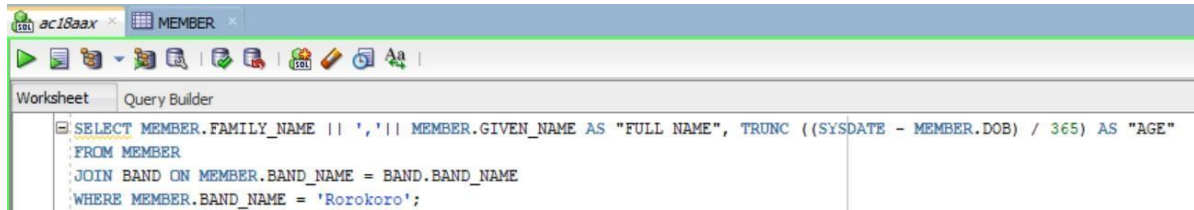
```

Name          Null?      Type
-----
BAND_NAME     NOT NULL  VARCHAR2 (30)
AGENT_NO      NUMBER
DESCRIPTION   VARCHAR2 (40)
Name          Null?      Type
-----
MEMBER_ID     NOT NULL  NUMBER
GIVEN_NAME    NOT NULL  VARCHAR2 (20)
FAMILY_NAME   NOT NULL  VARCHAR2 (20)
DOB           NOT NULL  DATE
JOB_TYPE      NOT NULL  VARCHAR2 (20)
BAND_NAME     NOT NULL  VARCHAR2 (20)

```

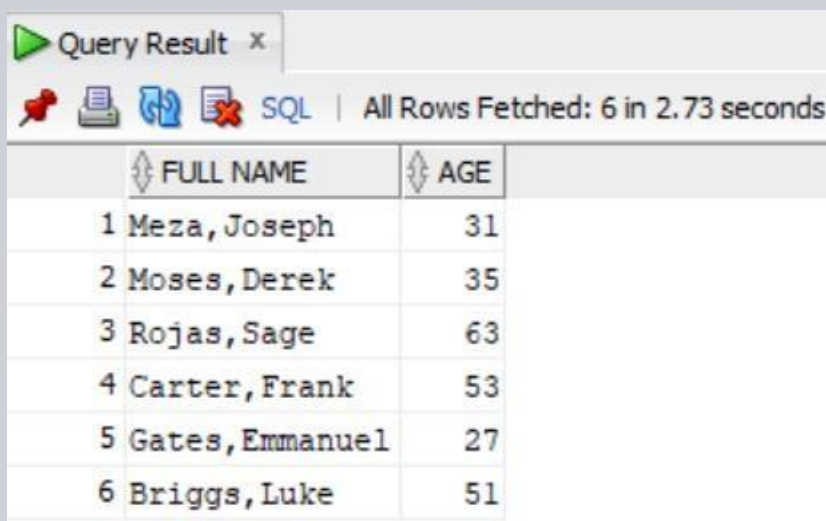

9. Report the name and age of the band members in Rorokoro

Screenshot of query:



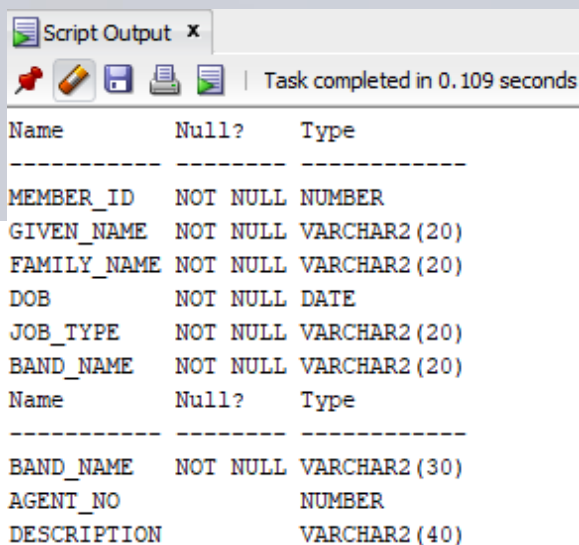
```
SELECT MEMBER.FAMILY_NAME || ', ' || MEMBER.GIVEN_NAME AS "FULL NAME", TRUNC ((SYSDATE - MEMBER.DOB) / 365) AS "AGE"
FROM MEMBER
JOIN BAND ON MEMBER.BAND_NAME = BAND.BAND_NAME
WHERE MEMBER.BAND_NAME = 'Rorokoro';
```

Screenshot of output:



	FULL NAME	AGE
1	Meza, Joseph	31
2	Moses, Derek	35
3	Rojas, Sage	63
4	Carter, Frank	53
5	Gates, Emmanuel	27
6	Briggs, Luke	51

Table Description:



Name	Null?	Type
MEMBER_ID	NOT NULL	NUMBER
GIVEN_NAME	NOT NULL	VARCHAR2 (20)
FAMILY_NAME	NOT NULL	VARCHAR2 (20)
DOB	NOT NULL	DATE
JOB_TYPE	NOT NULL	VARCHAR2 (20)
BAND_NAME	NOT NULL	VARCHAR2 (20)

Name	Null?	Type
BAND_NAME	NOT NULL	VARCHAR2 (30)
AGENT_NO		NUMBER
DESCRIPTION		VARCHAR2 (40)

10. Create a playlist ordered by stage name then date, then time for all performances. List them by stage name, band name, DAY (such as Friday, Saturday etc.) and Time of performance

Screenshot of query:

```

SELECT STAGE.STAGE_NAME, PERFORMANCE.BAND_NAME, TO_CHAR (PERFORMANCE.DATE_TIME, 'DAY MONTH YYYY HH:MI:SS') AS "DATE AND TIME"
FROM STAGE
JOIN PERFORMANCE ON STAGE.STAGE_NO = PERFORMANCE.STAGE_NO
ORDER BY STAGE.STAGE_NAME, PERFORMANCE.DATE_TIME;

```

Screenshot of output:

	STAGE_NAME	BAND_NAME	DATE AND TIME
1	Glade	Rorokoro	SATURDAY DECEMBER 2019 10:00:00
2	Pyramid	Outkast	TUESDAY DECEMBER 2019 07:00:00
3	Pyramid	The Animals	WEDNESDAY JANUARY 2020 09:00:00
4	Sonic	Castille	MONDAY DECEMBER 2019 02:00:00
5	Spartan	Blue Pears	MONDAY JANUARY 2020 08:30:00
6	StoneBridge	Roots	FRIDAY DECEMBER 2019 01:00:00

Table Description:

```

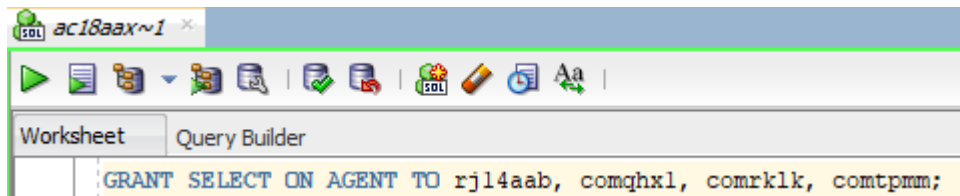
Name          Null?      Type
-----
STAGE_NO      NOT NULL  NUMBER
STAGE_NAME    VARCHAR2(20)
CAPACITY      NUMBER

Name          Null?      Type
-----
PERFORMANCE_ID NOT NULL  NUMBER
STAGE_NO      NUMBER
BAND_NAME     VARCHAR2(20)
DATE_TIME     TIMESTAMP(6)

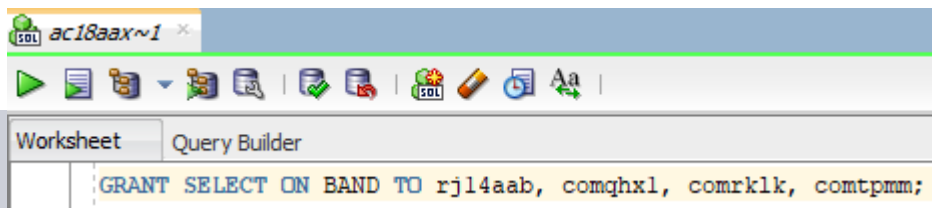
```

Screenshot of granted access:

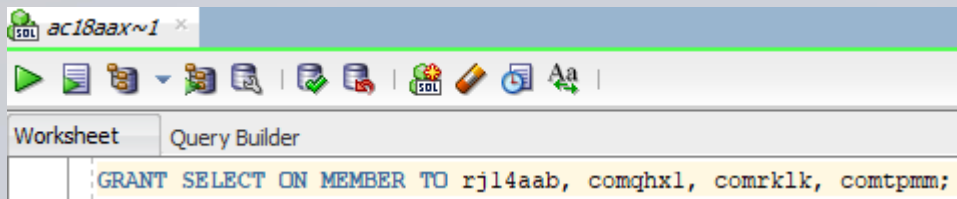
AGENT:



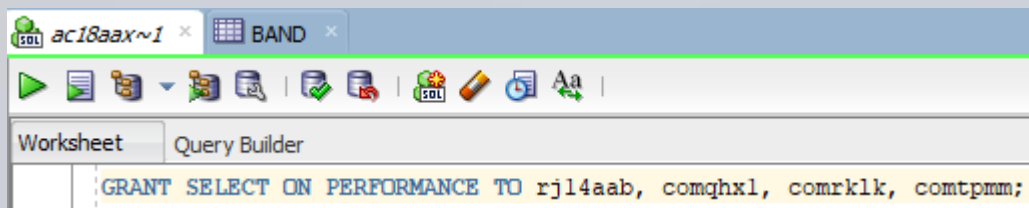
BAND:



MEMBER:



PERFORMANCE:



STAGE:

