INFS2200 Assignment

Roy Portas - 43560846

Task 1: Constraints

Part A

```sql
select * from USER_CONSTRAINTS;
```

Output:

```
OWNER                                                                  CONSTRAINT_NAME          C TABLE_NAME
SEARCH_CONDITION
R_OWNER                                                                R_CONSTRAINT_NAME        DELETE_RU STATUS   DEFERRABLE     DEFERRED
VALIDATED     GENERATED     BAD RELY LAST_CHAN INDEX_OWNER             INDEX_NAME               INVALID VIEW_RELATED
S4356084                                                               PK_BIRD_ID               P BIRDS

                            ENABLED  NOT DEFERRABLE IMMEDIATE
VALIDATED     USER NAME         20/OCT/16 S4356084                     PK_BIRD_ID

OWNER                                                                  CONSTRAINT_NAME          C TABLE_NAME
SEARCH_CONDITION
R_OWNER                                                                R_CONSTRAINT_NAME        DELETE_RU STATUS   DEFERRABLE     DEFERRED
VALIDATED     GENERATED     BAD RELY LAST_CHAN INDEX_OWNER             INDEX_NAME               INVALID VIEW_RELATED
S4356084                                                               PK_ORGANISATION_ID       P ORGANISATIONS

                            ENABLED  NOT DEFERRABLE IMMEDIATE
VALIDATED     USER NAME         20/OCT/16 S4356084                     PK_ORGANISATION_ID

OWNER                                                                  CONSTRAINT_NAME          C TABLE_NAME
SEARCH_CONDITION
R_OWNER                                                                R_CONSTRAINT_NAME        DELETE_RU STATUS   DEFERRABLE     DEFERRED
VALIDATED     GENERATED     BAD RELY LAST_CHAN INDEX_OWNER             INDEX_NAME               INVALID VIEW_RELATED
S4356084                                                               PK_SPOTTER_ID            P SPOTTERS

                            ENABLED  NOT DEFERRABLE IMMEDIATE
VALIDATED     USER NAME         20/OCT/16 S4356084                     PK_SPOTTER_ID
```

Part B

```sql
alter table SPOTTERS
add constraint FK_ORG_ID_TO_ORG_ID
foreign key (organisation_id) references ORGANISATIONS (organisation_id);

alter table SIGHTINGS
add constraint PK_SIGHTING_ID
PRIMARY KEY (sighting_id);

alter table SIGHTINGS
add constraint FK_SPOTTER_ID_TO_SPOTTER_ID
foreign key (spotter_id) references SPOTTERS (spotter_id);

alter table SIGHTINGS
add constraint FK_BIRD_ID_TO_BIRD_ID
foreign key (bird_id) references BIRDS (bird_id);

alter table ORGANISATIONS
modify ORGANISATION_NAME
constraint NN_ORGANISATION_NAME NOT NULL;

alter table SPOTTERS
```

```
modify SPOTTER_NAME
constraint NN_SPOTTER_NAME NOT NULL;

alter table SIGHTINGS
add constraint CK_SIGHTING_DATE
check (SIGHTING_DATE <= TO_DATE('2016-12-31', 'YYYY-MM-DD'));
```

Output:

```
Table altered.


Table altered.


Table altered.


Table altered.


Table altered.


Table altered.


Table altered.
```

Task 2: Triggers

Part A

```
create sequence SEQ_SIGHTINGS
start with 300000
increment by 1;

create or replace trigger TR_SIGHTING_ID
before insert on "SIGHTINGS"
for each row
begin
    select "SEQ_SIGHTINGS".NEXTVAL into :NEW.sighting_id from DUAL;
end;
/
```

Output:

```
Sequence created.

Trigger created.
```

Part B

```
create or replace trigger TR_SIGHTING_DESC
before insert on "SIGHTINGS"
for each row
```

```
begin
    if :NEW.latitude < -28.4 then
        /* Its less than the middle latitude, thus south */
        if :NEW.longitude < 151.25 then
            /* Its less than the middle longitude */
            select CONCAT(CONCAT('A bird of the species ', BIRD_NAME),
                          ' was spotted in the south-west part of the observation area' )
            into :NEW.DESCRIPTION from DUAL
            inner join BIRDS
            on BIRDS.BIRD_ID = :NEW.BIRD_ID;
        else
            /* Its greater than the middle longitude */
            select CONCAT(CONCAT('A bird of the species ', BIRD_NAME),
                          ' was spotted in the south-east part of the observation area' )
            into :NEW.DESCRIPTION from DUAL
            inner join BIRDS
            on BIRDS.BIRD_ID = :NEW.BIRD_ID;
        end if;
    else
        /* Its greater than the middle latitude */
        if :NEW.longitude < 151.25 then
            /* Its less than the middle longitude */
            select CONCAT(CONCAT('A bird of the species ', BIRD_NAME),
                          ' was spotted in the north-west part of the observation area' )
            into :NEW.DESCRIPTION from DUAL
            inner join BIRDS
            on BIRDS.BIRD_ID = :NEW.BIRD_ID;
        else
            /* Its greater than the middle longitude */
            select CONCAT(CONCAT('A bird of the species ', BIRD_NAME),
                          ' was spotted in the north-east part of the observation area' )
            into :NEW.DESCRIPTION from DUAL
            inner join BIRDS
            on BIRDS.BIRD_ID = :NEW.BIRD_ID;
        end if;
    end if;

end;
/
```

Output:

```
Trigger created.
```

Part C

```
INSERT INTO sightings (spotter_id, bird_id, latitude,
longitude, sighting_date)
VALUES (2457, 901, -28.0, 152, '09-MAR-2016');
INSERT INTO sightings (spotter_id, bird_id, latitude,
longitude, sighting_date)
VALUES (1024, 512, -25.6, 153, '09-MAR-2016');
```

Output:

```
1 row created.
```

```
1 row created.
```

Part D

```sql
select * from sightings where sighting_date = '09-MAR-2016';
```

Output:

```
SIGHTING_ID SPOTTER_ID    BIRD_ID   LATITUDE  LONGITUDE SIGHTING_
----------- ---------- ---------- ---------- ---------- ---------
DESCRIPTION
--------------------------------------------------------------------------------------------------------
     300000       2457        901        -28        152 09/MAR/16
A bird of the species Australian pied cormorant was spotted in the north-east part of the observation area

     300001       1024        512      -25.6        153 09/MAR/16
A bird of the species Mrs. Humes pheasant was spotted in the north-east part of the observation area
```

Task 3: Views

Part A

```sql
create or replace view V_ORGANISATION_BIRD_COUNT as select org.ORGANISATION_NAME, count(*) "bird_count"
from ORGANISATIONS org
inner join SPOTTERS sp
    on org.ORGANISATION_ID = sp.ORGANISATION_ID
inner join SIGHTINGS si
    on sp.SPOTTER_ID = si.SPOTTER_ID
group by ORGANISATION_NAME;
```

Output:

```
View created.
```

Part B

```sql
create materialized view MV_ORGANISATION_BIRD_COUNT as
select org.ORGANISATION_NAME, count(*) "bird_count"
from ORGANISATIONS org
inner join SPOTTERS sp
    on org.ORGANISATION_ID = sp.ORGANISATION_ID
inner join SIGHTINGS si
    on sp.SPOTTER_ID = si.SPOTTER_ID
group by ORGANISATION_NAME;
```

Output:

```
Materialized view created.
```

Part C

```sql
SELECT * FROM V_ORGANISATION_BIRD_COUNT;
```

```
ORGANISATION_NAME                                                  bird_count
------------------------------------------------------------------ ----------
Greenpeace                                                              33901
Department of Environmental Sciences                                    34457
Environmental Protection Agency                                         33195
Peoples Association for the Conservation of the Environment             34885
National Bird Observatory                                               32469
Royal Society for the Protection of Birds                               32899
National Bird Spotting Association                                      32792
Highlands Bird Watching Society                                         33294

8 rows selected.

Elapsed: 00:00:00.12
```

```
SELECT * FROM MV_ORGANISATION_BIRD_COUNT;
```

```
ORGANISATION_NAME                                                  bird_count
------------------------------------------------------------------ ----------
Greenpeace                                                              33901
Department of Environmental Sciences                                    34457
Environmental Protection Agency                                         33195
Peoples Association for the Conservation of the Environment             34885
National Bird Observatory                                               32469
Royal Society for the Protection of Birds                               32899
National Bird Spotting Association                                      32792
Highlands Bird Watching Society                                         33294

8 rows selected.

Elapsed: 00:00:00.05
```

The materialized view is much faster because the data is stored in a table, instead of having to be queried through the other tables.

Task 4: Function Based Indexes

Part A

```
select * from (
    select SIGHTING_ID, sqrt(power((LATITUDE + -28), 2) + power((LONGITUDE + 151), 2))
    as DISTANCE
    from SIGHTINGS
    order by DISTANCE desc
) where ROWNUM <= 1;
```

```
SIGHTING_ID   DISTANCE
----------- ----------
       5139 309.177765

Elapsed: 00:00:00.20
```

Part B

```
create index IDX_HEADQUARTERS_DISTANCE on SIGHTINGS(sqrt(power((LATITUDE + -28), 2) + power((LONGITUDE + 1
commit;

analyze INDEX IDX_HEADQUARTERS_DISTANCE VALIDATE STRUCTURE;
commit;
```

Part C

```
select * from (
    select SIGHTING_ID, sqrt(power((LATITUDE + -28), 2) + power((LONGITUDE + 151), 2))
    as DISTANCE
    from SIGHTINGS
    order by DISTANCE desc
) where ROWNUM <= 1;
```

```
SIGHTING_ID   DISTANCE
----------- ----------
       5139 309.177765

Elapsed: 00:00:00.18
SQL>
```

The index will be indexing the queries, so when the database goes to calculate the distance, it will first look up the equation in the index, and return the precomputed value if found.

We don't get massive performance boosts because a lot of the distances are unique. However if there was many duplicates, such as birds at the exact same position, there would be more noticable improvements.

Task 5: Execution Plan and Analysis

Part A

```
explain plan for select SIGHTING_ID, SPOTTER_NAME, SIGHTING_DATE
from SIGHTINGS
inner join SPOTTERS
    on SPOTTERS.SPOTTER_ID = SIGHTINGS.SPOTTER_ID
where SIGHTINGS.SPOTTER_ID = 1255;
```

Running the above command yields to following result

```
SIGHTING_ID SPOTTER_NAME                                           SIGHTING_
----------- ---------------------------------------------------- ---------
     225748 Alana Debari                                         12/MAR/01
     242674 Alana Debari                                         17/NOV/01
     241448 Alana Debari                                         28/JUN/00
     247606 Alana Debari                                         21/OCT/14
     248257 Alana Debari                                         20/JUL/15
     249309 Alana Debari                                         04/APR/15
     247582 Alana Debari                                         25/DEC/10
     263323 Alana Debari                                         13/JUN/05

52 rows selected.

Elapsed: 00:00:00.19
```

```
SELECT PLAN_TABLE_OUTPUT FROM TABLE (DBMS_XPLAN.DISPLAY);
```

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------------------
Plan hash value: 4071757951

-----------------------------------------------------------------------------------------------
| Id  | Operation                     | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |               |    82 |  6150 |  1402   (1)| 00:00:17 |
|   1 |  NESTED LOOPS                 |               |    82 |  6150 |  1402   (1)| 00:00:17 |
|   2 |   TABLE ACCESS BY INDEX ROWID| SPOTTERS      |     1 |    40 |     0   (0)| 00:00:01 |
|*  3 |    INDEX UNIQUE SCAN          | PK_SPOTTER_ID |     1 |       |     0   (0)| 00:00:01 |
|*  4 |   TABLE ACCESS FULL           | SIGHTINGS     |    82 |  2870 |  1402   (1)| 00:00:17 |
-----------------------------------------------------------------------------------------------

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   3 - access("SPOTTERS"."SPOTTER_ID"=1255)
   4 - filter("SIGHTINGS"."SPOTTER_ID"=1255)

Note
-----
   - dynamic sampling used for this statement (level=2)

21 rows selected.

Elapsed: 00:00:00.59
```

The query plan can be described as a series of steps:

1. Run a index scan on the `SPOTTER_ID` in the `SPOTTERS` table to evaluate the where clause
2. Rows are located by the `ROWID` index in the `SPOTTERS` table and the the entire `SIGHTINGS` table is read
3. The `SPOTTERS` table becomes the outer loop and the `SIGHTINGS` table becomes the inner loop, it then joins the tables with a nested loop
4. Apply the select statement on the result

Part B

```
alter table SIGHTINGS
drop constraint FK_SPOTTER_ID_TO_SPOTTER_ID;

alter table SPOTTERS
drop constraint PK_SPOTTER_ID;

drop index PK_SPOTTER_ID;

explain plan for select SIGHTING_ID, SPOTTER_NAME, SIGHTING_DATE
from SIGHTINGS
inner join SPOTTERS
    on SPOTTERS.SPOTTER_ID = SIGHTINGS.SPOTTER_ID
where SIGHTINGS.SPOTTER_ID = 1255;

SELECT PLAN_TABLE_OUTPUT FROM TABLE (DBMS_XPLAN.DISPLAY);
```

Running the above command yields the following time

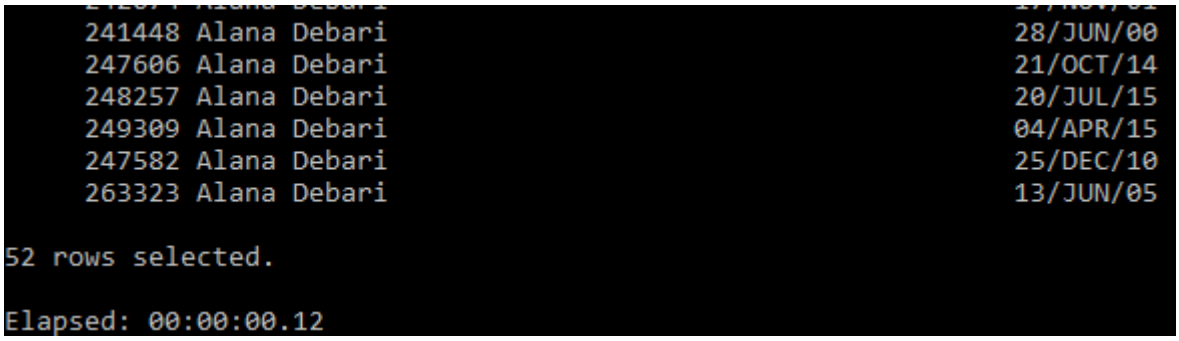

Figure 1: Part 2 timing

With the primary key of spotter removed, the database is unable to do a unique scan through the table, instead it needs to do a HASH join, which creates a hash table on one of the tables and scans the other table, matching between the hash table and other table.

This method is actually faster to execute, because the query optimizer will filter both tables by `SPOTTER_ID`, which means the size of both tables would be small. This is because the `SPOTTERS` table should only have a single match to the given `SPOTTER_ID`, assuming the data has not been changed since the primary key has been dropped.

The other table will only contain the number of spottings the given `SPOTTER_ID` has, which is a very small subset of the entire table. Then both tables are joined, which is quick since both tables are small.

Part C

```
analyze index PK_BIRD_ID validate structure offline;
```

Tree Height:

```
select HEIGHT from INDEX_STATS;
```

Tree height is **2**

Leaf Blocks:

```
select LF_BLKS from INDEX_STATS;
```

9 leaf blocks

Block accesses needed for a direct full scan of the BIRDS table:

```
select LF_BLKS from INDEX_STATS;
select BR_BLKS from INDEX_STATS;
```

```
= Number of leaf blocks + Number of branch blocks
= 9 + 1
= 10
```