TASK 5 REPORT

I used my 3NF schema for this task

1. NESTED LOOP JOIN

EXPLAIN PLAN FOR

Select /*+ USE_NL(cs c) */ *

From ce_channel_classes cs

inner join ce_channels c

on c.channel_class_id=cs.channel_class_id;

SELECT * FROM table (DBMS_XPLAN.DISPLAY);

I	d	I	Operation Name	I	Rows	I	Bytes	I	Cost (%CI	PU)	T	ime	I
 	0	1	SELECT STATEMENT	 I	9	1	1080	1	9	(0)		0:00:01	1
I	1	Ī	NESTED LOOPS	-	9	ı	1080	I	9	(0)	0	0:00:01	I
I	2	Ī	TABLE ACCESS FULL CE_CHANNEL_CLASSES	-1	3	ı	174	1	3	(0)	0	0:00:01	I
*	3	ī	TABLE ACCESS FULL CE CHANNELS	-	3	ı	186	ī	2	(0)	0	0:00:01	1

2. SORT-MERGE JOIN

EXPLAIN PLAN FOR

Select /*+ USE_MERGE(p pt)*/*

From ce_products p
inner join ce_product_types pt
on p.product_type_id=pt.product_type_id

Where p.unit_cost>1000;

SELECT * FROM table (DBMS XPLAN.DISPLAY):

Ic	i	0)peratio	on	I	Name	I	Rows	I	Bytes	TempSpc	Cost	(%CPU)	Time
	0	S	SELECT S	STATEMENT				4997	1	780KI	I	189	(2)	00:00:0
	1		MERGE 0	JOIN	I		1	4997	1	780K	1	189	(2)	00:00:0
	2		TABLE	ACCESS BY	INDEX ROWID	CE_PRODUCT_TYP	PES	123	1	7626	1	4	(0)	00:00:0
	3		INDEX	K FULL SCA	an I	CE_PRODUCT_TYP	PE_PK	123	1	I	1	1	(0)	00:00:0
k	4		SORT (JOIN	I		1	4997	1	478K	1240K	185	(2)	00:00:0
k	5		TABLE	E ACCESS E	TULL	CE PRODUCTS	1	4997	1	478K	1	68	(0)	00:00:0

3: HASH JOIN

EXPLAIN PLAN FOR Select /*+ USE_HASH(p pt)*/* From ce_sales s inner join ce_products p

on p.product_id=s.product_id;

SELECT * FROM table (DBMS_XPLAN.DISPLAY);

Pla	n l	has	sh value: 32426720	086									
I	d	ı	Operation	N	ame	ı	Rows	Bytes	TempSpc	Cost	(%CPU)	Time	
I	0	1	SELECT STATEMENT	1		1	1025K	187M	1 1	9157	(1)	00:00:01	L
*	1	1	HASH JOIN	- 1		1	1025K	187M	1152K	9157	(1)	00:00:01	L
I	2	1	TABLE ACCESS FO	JLL C	E_PRODUCTS	1	10666	1020K	1 1	68	(0)	00:00:01	L
I	3	1	TABLE ACCESS FO	JLL C	E_SALES	1	1025K	91M	1	3878	(1)	00:00:01	L
Pre	dio	cat	te Information (io	dentif:	ied by ope:	rat	tion id)	:					
	1 -	- ē	access("P"."PRODUC	CT_ID":	="S"."PROD	JC.	r_ID")						

4. CARTESIAN JOIN

ORDERED HINT AND WRONG PREDICATE ORDER OR JOIN TO THE TABLE WITH 0 ROWS.

Select ce_channel_classes.channel_class, ce_channels.channel_description, ce_channels.channel_id from ce_channel_classes

EXPLAIN PLAN FOR

Select /*+ ORDERED*/ *

From ce_channel_classes ,ce_channels

where ce_channel_classes.channel_class_id=ce_channels.channel_class_id; and ce_channel_classes.channel_class='tDirect';

SELECT * FROM table (DBMS_XPLAN.DISPLAY);

1	P1	an	ha	sh	value	: 10280	96691											
2																		
3																		
4	I	Id	- 1	0	perati	on		Ī	Name	1	Rows	Ī	Bytes	1	Cost	(%CPU)	Time	- 1
5																		
6	L	0	1	S	ELECT :	STATEME	INT	1		1	27	1	540	1	9	(0)	00:00:0	1
7	L	1	. 1		MERGE	JOIN C	ARTESIA	N I		1	27	I	540	1	9	(0)	00:00:0	1
8	L	2	1		TABLE	ACCESS	FULL	1	CE_CHANNEL_CLASSES	1	3	1	21	1	3	(0)	00:00:0	1
9	L	3	1		BUFFE	R SORT		1		1	9	1	117	1	6	(0)	00:00:0	1
10	L	4	ı		TABL	E ACCES	S FULL	Ī	CE_CHANNELS	1	9	I	117	1	2	(0)	00:00:0	1
11																		

W/O JOIN CONDITIONS.

EXPLAIN PLAN FOR

Select ce_channel_classes.channel_class, ce_channels.channel_description, ce_channels.channel_id

From ce_channel_classes ,ce_channels;

SELECT * FROM table (DBMS_XPLAN.DISPLAY);

1	P	la	n	ha	sh value:	1028096	691												
2																			
3	-																		-
4	ı	I	d	1	Operation	n	- 1	Name		ĺ	Rows	1	Bytes	Ī	Cost	(%CPU)	Time		1
5	-																		-
6	1		0	-	SELECT S	STATEMENT	- 1			Ī	27	1	540	Ī	9	(0)	00:00	:01	1
7	1		1	-	MERGE J	JOIN CART	ESIAN			Ī	27	1	540	Ī	9	(0)	00:00	:01	1
8	1		2	-	TABLE	ACCESS F	ULL	CE_CHANNEL_	CLASSES	Ī	3	1	21	Ī	3	(0)	00:00	:01	1
9	I		3	-	BUFFER	SORT	- 1			Ī	9	1	117	Ī	6	(0)	00:00	:01	1
10	1		4	1	TABLE	ACCESS	FULL	CE_CHANNELS	3	I	9	1	117	Ī	2	(0)	00:00	:01	1
11	-																		-

5. LEFT/RIGHT OUTER JOINS

ANSI SYNTAX (LEFT/RIGHT JOIN)

EXPLAIN PLAN FOR

Select *

From ce_products p

left outer join ce_product_types pt

on p.product_type_id=pt.product_type_id;

SELECT * FROM table (DBMS_XPLAN.DISPLAY);

1			_	FABLE_OU sh value:		29812										
2				on varac.	. 5/1420	33012										
3			_								_					_
4]	d	I	Operation	on		1	Name	- 1	Rows	1	Bytes	Cost	(%CPU)	Time	
5																-
6	1	0	Ī	SELECT S	STATEMEN	T	Ī		-1	10666	Ī	1666K	71	(0)	00:00:01	
7	1*	1	Ī	HASH JO	OIN RIGH	IT OUTER	RΙ		-1	10666	Ī	1666K	71	(0)	00:00:01	
8	1	2	Ī	TABLE	ACCESS	FULL	Ī	CE_PRODUCT_TYPES	1	123	Ī	7626	3	(0)	00:00:01	
9	1	3	Ī	TABLE	ACCESS	FULL	1	CE_PRODUCTS	-1	10666	Ī	1020K	68	(0)	00:00:01	
10																-
11																
12	Dane			. Tofou				ed by operation i	-23							

ORACLE SYNTAX (+)

EXPLAIN PLAN FOR

Select *

From ce_products p, ce_product_types pt

Where p.product_type_id=pt.product_type_id(+);

SELECT * FROM table (DBMS_XPLAN.DISPLAY);

	♦ PLAN_TABLE_OUTPUT
1	Plan hash value: 3714289812
2	
3	
4	Id Operation
5	
6	0 SELECT STATEMENT 10666 1666K 71 (0) 00:00:01
7	* 1 HASH JOIN RIGHT OUTER 10666 1666K 71 (0) 00:00:01
8	2 TABLE ACCESS FULL CE_PRODUCT_TYPES 123 7626 3 (0) 00:00:01
9	3 TABLE ACCESS FULL CE_PRODUCTS 10666 1020K 68 (0) 00:00:01
10	
11	
12	Predicate Information (identified by operation id):
13	
14	
15	1 - access("P"."PRODUCT_TYPE_ID"="PT"."PRODUCT_TYPE_ID"(+))

6. FULL OUTER JOIN

ANSI SYNTAX (OUTER JOIN)

EXPLAIN PLAN FOR

Select count(*)

From ce_products p

full outer join ce_brands b

on p.product_brand_id=b.product_brand_id;

SELECT * FROM table (DBMS_XPLAN.DISPLAY);



			_	TABLE_OUTPUT											_
1	Pla	n l	as	sh value: 2930694060											
2															
3															
4	I	d	1	Operation		Name	1	Rows	Ī	Bytes	Ī	Cost	(%CPU)	Time	
5															
6	L	0	1	SELECT STATEMENT	1		1	1	Ī		Ī	70	(0)	00:00:01	
7	L	1	1	SORT AGGREGATE	1		1	1	1		Ī		- 1		
8	L	2	1	VIEW	1.3	VW_FOJ_0	1	10666	Ī		Ī	70	(0)	00:00:01	
9	1*	3	1	HASH JOIN FULL OUTER	1		1	10666	1	85328	1	70	(0)	00:00:01	
10	L	4	1	INDEX FAST FULL SCA	NI (CE_PRODUCT_BRAND_PK	1	424	1	1696	Ī	2	(0)	00:00:01	
11	I	5	ī	TABLE ACCESS FULL	1 (CE_PRODUCTS	1	10666	Ī	42664	Ī	68	(0)	00:00:01	
12															
13															
14	Pre	dio	at	e Information (identifie	ed I	by operation id):									
15															

ORACLE SYNTAX (+)

EXPLAIN PLAN FOR

Select count (*)

```
From (
Select *
From ce_products p, ce_brands b
Where p.product_brand_id=b.product_brand_id(+)
union
Select *
From ce_products p, ce_brands b
Where p.product_brand_id(+)=b.product_brand_id);
SELECT * FROM table (DBMS_XPLAN.DISPLAY);
```

1	10666

1	Plan hash	value: 541405082	2							
2										
3 -										
4	Id C	peration	[]	Name	Rows	-1	Bytes	Cost	(%CPU) :	Time
5 -										
6	0 5	ELECT STATEMENT	1		1	L	- 1	1680	(1)	00:00:01
7	1	SORT AGGREGATE	1		1	L	- 1		1	ı
8	2	VIEW	1		21332	2	- 1	1680	(1)	00:00:01
9	3	HASH UNIQUE	I		21332	2	3374K	1680	(1)	00:00:01
10	4	UNION-ALL	1		1	- 1	- 1		1	ı
11	* 5	HASH JOIN RIG	HT OUTER		10666	5	1687K	72	(2)	00:00:01
12	6	TABLE ACCESS	FULL	CE_BRANDS	424	1	27136	3	(0)	00:00:01
13	7	TABLE ACCESS	FULL	CE_PRODUCTS	10666	5	1020K	68	(0)	00:00:01
14	* 8	HASH JOIN OUT	TER		10666	5	1687K	72	(2)	00:00:01
15	9	TABLE ACCESS	FULL	CE_BRANDS	424	1	27136	3	(0)	00:00:01
16	10	TABLE ACCESS	FULL	CE_PRODUCTS	10666	5	1020K	68	(0)	00:00:01
17										
18										
19	Predicate	Information (ide	ntified by	operation :	id):					
20										
21										
22	5 - ac	cess("P"."PRODUC	BRAND ID"	="B"."PRODUC	CT BRANI) II	0" (+))			
23		cess("P"."PRODUC								

7. MERGE

```
MERGE INTO ce_products ch

USING (SELECT DISTINCT

c.product_id AS product_srcid,

'personnel_sales' AS product_source_system,

'src_products' AS product_source_entity,

COALESCE(c.product_name,'N/A') AS product,

COALESCE(c.unit_price,'-1') AS unit_price,

COALESCE(c.unit_cost,'-1') AS unit_cost,

COALESCE(c2.product_brand_id,-1) AS product_brand_id,

COALESCE(c3.product_type_id,-1) AS product_type_id
```

FROM SA_SOURCE_SYSTEM_1.src_products c LEFT JOIN bl_3nf.ce_brands c2

ON c.brand_id=c2.product_brand_srcid

LEFT JOIN bl_3nf.ce_product_types c3

on c.product_type_id=c3.product_type_srcid) t

ON (ch.product_srcid = t.product_srcid

AND ch.product_source_system = t.product_source_system

AND ch.product_source_entity = t.product_source_entity)

WHEN MATCHED THEN

Update set

product=product, unit_price=unit_price, unit_cost=unit_cost,

product_brand_id=product_brand_id, product_type_id=product_type_id,ta_update_dt=SYSDATE WHEN NOT MATCHED THEN

INSERT(product_id, product_srcid, product_source_system, product_source_entity, product, unit_price, unit_cost, product_brand_id, product_type_id,ta_update_dt, ta_insert_dt) VALUES(seq_ce_product_id.NEXTVAL,t.product_srcid, t.product_source_system, t.product_source_entity, t.product, t.unit_price, t.unit_cost, t.product_brand_id, t.product_type_id,SYSDATE,SYSDATE);

8. A LIST OF COMBINATION OF DIFFERENT TABLES

Table "A"	Table "B"	Join type description
Small Table w/o index on join field	Small Table w/o index on join field	Nested Loops
Small Table w/o index on join field	Small Table with index on join field	Nested Loops
Small Table w/o index on join field	Big Table with index on join field	Nested Loops
Small Table w/o index on join field	Big Table w/o index on join field	Hash Join
Big table, but can fit comletely in PDA	Big Table w/o index on join field	Hash Join
Big table, but cannot fit comletely in PDA	Big Table w/o index on join field	Sort Merge Join
Big table, use condition < >	Big table	Sort Merge Join
Big table ordered by join column	Big table ordered by join column	Sort Merge Join
Big table	0 rows table	Cartesian Join
Big table	Big table	Cartesian only if we use CROSS JOIN