

Advanced Database

TIS3351

TC101/TC201

Data Warehouse for Eco Used Cars

Student ID	Student Name	Tutorial Section
1092700670	MOHAMMAD MASOUDIMOGHADDAM	TT01
1111113245	MOHAMED SALEH	TT01

Contents

1	Scenario	2
2	Star Schema	3
3	Calculation for the fact table size and storage in the database schema	4
4	SQL commands for Data Definition Language (DDL)	4
5	Queries	7
6	Pivot tables and graphs based on Views	10
7	Operational Database design and Data Warehouse	11

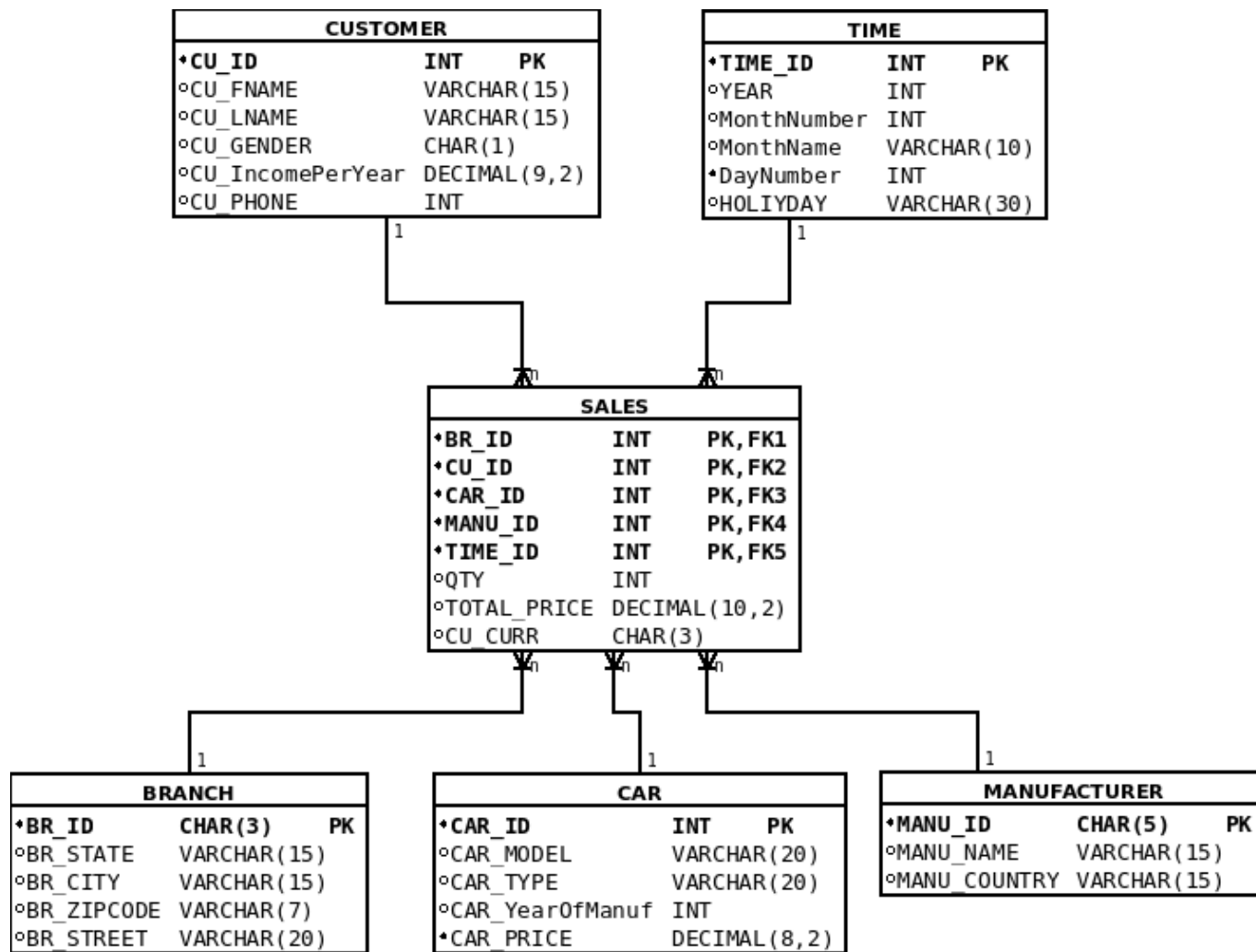
1 Scenario

Eco Used Cars is one of the leading car dealers in Malaysia. They have several branches across the states. The data warehouse for Eco Used Cars is used to track the price of used cars over the years. This data warehouse is intended to keep track of the following:

- Sales of used cars by model, manufacturer, year and region.
- Demographics of people who buy used cars (age, income, location).

This data warehouse will be used to predict the resale value of used cars, as well as the market group that should be targeted. Design and implement this data warehouse.

2 Star Schema



3 Calculation for the fact table size and storage in the database schema

Fact table storage = Number of Attributes * number of Records * filed size(Bytes)

Fact table: SALES table storage = $8 * 49 * 4$ bytes = 1568 bytes

4 SQL commands for Data Definition Language (DDL)

```
1 CREATE DATABASE DDL;
2 CONNECT TO DDL;
3
4
5 CREATE TABLE CAR (
6     CAR_ID INT NOT NULL GENERATED ALWAYS AS IDENTITY (START WITH 11111, INCREMENT BY 10),
7     CAR_MODEL VARCHAR(20),
8     CAR_TYPE VARCHAR(20),
9     CAR_YearOfManuf INT,
10    CAR_PRICE DECIMAL(8,2),
11    PRIMARY KEY (CAR_ID)
12 );
13
14 INSERT INTO CAR VALUES(DEFAULT, 'PERODUA MYVI', 'GENERAL', 2009, 25000);
15 INSERT INTO CAR VALUES(DEFAULT, 'PERODUA ALZA', 'MPVs', 2012, 30000.00);
16 INSERT INTO CAR VALUES(DEFAULT, 'PERODUA ALZA', 'MPVs', 2012, 30000.00);
17 INSERT INTO CAR VALUES(DEFAULT, 'PERODUA ALZA', 'MPVs', 2012, 30000.00);
18 INSERT INTO CAR VALUES(DEFAULT, 'PERODUA KANCIL', 'GENERAL', 2008, 5000);
19 INSERT INTO CAR VALUES(DEFAULT, 'PERODUA KANCIL', 'GENERAL', 2007, 5000);
20 INSERT INTO CAR VALUES(DEFAULT, 'PROTON SAGA', 'GENERAL', 2006, 8000);
21 INSERT INTO CAR VALUES(DEFAULT, 'PROTON SAGA', 'GENERAL', 2007, 8000);
22 INSERT INTO CAR VALUES(DEFAULT, 'PROTON WIRA', 'GENERAL', 2009, 12000);
23 INSERT INTO CAR VALUES(DEFAULT, 'PROTON WIRA', 'GENERAL', 2009, 12000);
24
25
26 CREATE TABLE MANUFACTURER (
27     MANU_ID CHAR(5) NOT NULL,
28     MANU_NAME VARCHAR(15),
29     MANU_COUNTRY VARCHAR(15),
30     PRIMARY KEY (MANU_ID)
31 );
32
33 INSERT INTO MANUFACTURER VALUES('JPE02', 'HONDA', 'JAPAN');
```

```

34 INSERT INTO MANUFACTURER VALUES( 'JPE01' , 'TOYOTA' , 'JAPAN' );
35 INSERT INTO MANUFACTURER VALUES( 'MAY01' , 'PROTON' , 'MALAYSIA' );
36 INSERT INTO MANUFACTURER VALUES( 'MAY02' , 'PERODUA' , 'MALAYSIA' );
37 INSERT INTO MANUFACTURER VALUES( 'GER02' , 'MERCEDES' , 'GERMANY' );
38
39 CREATE TABLE BRANCH (
40     BR_ID CHAR(3) NOT NULL,
41     BR.STATE VARCHAR(15) ,
42     BR.CITY VARCHAR(15) ,
43     BR.ZIPCODE VARCHAR(7) ,
44     BR.STREET VARCHAR(20) ,
45     PRIMARY KEY (BR_ID)
46 );
47
48 INSERT INTO BRANCH VALUES( 'B01' , 'SELANGOR' , 'KLANG' , '41200' , 'Jalan Batu Nilam 9' );
49 INSERT INTO BRANCH VALUES( 'B02' , 'PENANG' , 'GEORGE TOWN' , '10000' , 'Jalan Penang' );
50 INSERT INTO BRANCH VALUES( 'B03' , 'PAHANG' , 'KUANTAN' , '25000' , 'Jalan Besar' );
51 INSERT INTO BRANCH VALUES( 'B04' , 'MALACCA' , 'MELAKA' , '73300' , 'Batang Melaka Estate' );
52 INSERT INTO BRANCH VALUES( 'B05' , 'PERAK' , 'IPOH' , '30000' , 'Jalan Bandar Timah' );
53
54 CREATE TABLE CUSTOMER (
55     CU_ID INT NOT NULL GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),
56     CU.FNAME VARCHAR(15) ,
57     CU.LNAME VARCHAR(15) ,
58     CU.GENDER CHAR(1) CHECK((CU.GENDER = 'M') OR (CU.GENDER = 'F')) ,
59     CU.PHONE INT,
60     CU_IncomePerYear VARCHAR(15) , —DECIMAL(11,2)
61     CU.CURR CHAR(3) ,
62     PRIMARY KEY (CU_ID)
63 );
64
65 INSERT INTO CUSTOMER VALUES(DEFAULT, 'Sharon' , 'Edwards' , 'F' , '115754.50' , 130217292);
66 INSERT INTO CUSTOMER VALUES(DEFAULT, 'Kelly' , 'Baker' , 'F' , '14139.75' , 153933919);
67 INSERT INTO CUSTOMER VALUES(DEFAULT, 'Brandon' , 'Bradley' , 'M' , '120902.17' , 170571899);
68 INSERT INTO CUSTOMER VALUES(DEFAULT, 'Ruth' , 'Williams' , 'F' , '215095.72' , 177650041);
69 INSERT INTO CUSTOMER VALUES(DEFAULT, 'Clarence' , 'Mccoy' , 'M' , '272064.70' , 157177001);
70 INSERT INTO CUSTOMER VALUES(DEFAULT, 'Anne' , 'Hayes' , 'F' , '26111.89' , 167915545);
71 INSERT INTO CUSTOMER VALUES(DEFAULT, 'Norma' , 'Hawkins' , 'F' , '122914.26' , 170603072);
72 INSERT INTO CUSTOMER VALUES(DEFAULT, 'Jack' , 'Hughes' , 'M' , '157912.25' , 120143412);
73 INSERT INTO CUSTOMER VALUES(DEFAULT, 'Rachel' , 'Simmons' , 'F' , '24136.17' , 122987635);
74 INSERT INTO CUSTOMER VALUES(DEFAULT, 'Shirley' , 'Smith' , 'F' , '221846.04' , 130592077);
75
76
77
78 CREATE TABLE TIME (
79     TIME_ID INT NOT NULL GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),
80     TIME_YEAR INT WITH DEFAULT 2014,
81     MonthNumber INT CHECK(MonthNumber <= 12),
82     MonthName CHAR(10) ,

```

```

83 DayNumber INT CHECK(DayNumber <= 31),
84 HOLIDAY VARCHAR(30),
85 PRIMARY KEY (TIME_ID)
86 );
87
88 INSERT INTO TIME VALUES(DEFAULT, DEFAULT,1, 'JANUARY',29,NULL);
89 INSERT INTO TIME VALUES(DEFAULT, DEFAULT,1, 'JANUARY',30,NULL);
90 INSERT INTO TIME VALUES(DEFAULT, DEFAULT,1, 'JANUARY',31,'Chinese New Year(First day)');
91 INSERT INTO TIME VALUES(DEFAULT, DEFAULT,2, 'FEBRUARY',1,'Chinese New Year(Second day)');
92 INSERT INTO TIME VALUES(DEFAULT, DEFAULT,2, 'FEBRUARY',2,NULL);
93 INSERT INTO TIME VALUES(DEFAULT, DEFAULT,2, 'FEBRUARY',3,NULL);
94 INSERT INTO TIME VALUES(DEFAULT, DEFAULT,2, 'FEBRUARY',4,NULL);
95 INSERT INTO TIME VALUES(DEFAULT, DEFAULT,2, 'FEBRUARY',5,NULL);
96 INSERT INTO TIME VALUES(DEFAULT, DEFAULT,2, 'FEBRUARY',6,NULL);
97 INSERT INTO TIME VALUES(DEFAULT, DEFAULT,2, 'FEBRUARY',7,NULL);
98 INSERT INTO TIME VALUES(DEFAULT, DEFAULT,2, 'FEBRUARY',8,NULL);
99
100 CREATE TABLE SALES (
101 BR_ID CHAR(3) NOT NULL,
102 CAR_ID INT NOT NULL,
103 MANU_ID CHAR(5) NOT NULL,
104 CU_ID INT NOT NULL,
105 TIME_ID INT NOT NULL,
106 QTY INT WITH DEFAULT 1,
107 TOTAL_PRICE DECIMAL(9,2) WITH DEFAULT 0,
108 CU_Curr CHAR(3) WITH DEFAULT 'USD',
109 FOREIGN KEY (CAR_ID) REFERENCES CAR,
110 FOREIGN KEY (MANU_ID) REFERENCES MANUFACTURER,
111 FOREIGN KEY (BR_ID) REFERENCES BRANCH,
112 FOREIGN KEY (CU_ID) REFERENCES CUSTOMER,
113 FOREIGN KEY (TIME_ID) REFERENCES TIME,
114 PRIMARY KEY (CAR_ID,MANU_ID,BR_ID,CU_ID,TIME_ID)
115 );
116
117 INSERT INTO SALES VALUES('B01',11111,'MAY02',1,1,DEFAULT,DEFAULT);
118 INSERT INTO SALES VALUES('B01',11121,'MAY02',2,1,DEFAULT,DEFAULT);
119 INSERT INTO SALES VALUES('B01',11131,'MAY02',3,4,DEFAULT,DEFAULT);
120 INSERT INTO SALES VALUES('B01',11161,'MAY02',4,4,DEFAULT,DEFAULT);
121 INSERT INTO SALES VALUES('B01',11201,'MAY02',5,7,DEFAULT,DEFAULT);
122 INSERT INTO SALES VALUES('B01',11231,'MAY01',6,7,DEFAULT,DEFAULT);
123 INSERT INTO SALES VALUES('B01',11241,'MAY01',7,10,DEFAULT,DEFAULT);

```

5 Queries

```
1  ----- SALES AND MARKETING MANAGER -----
2  ---1)
3  CREATE TRIGGER CALCULATE_T.PRICE
4  BEFORE INSERT ON SALES
5  REFERENCING NEW AS N
6  FOR EACH ROW
7  MODE DB2SQL
8      SET TOTAL_PRICE = (SELECT PS.CAR_PRICE * N.QTY
9                          FROM CAR PS
10     WHERE N.CAR_ID = PS.CAR_ID);
11
12
13  ---2)
14  CREATE VIEW TOP_CAR.SALES AS
15  SELECT * FROM (SELECT CAR.MODEL, SUM(QTY) AS TOTALQTYSOLD, SUM(CAR.PRICE) AS
16     TOTAL_INCOME
17  FROM SALES S, CAR C
18  WHERE S.CAR_ID = C.CAR_ID
19  GROUP BY CAR.MODEL
20  ORDER BY TOTALQTYSOLD DESC);
21
22  ---3)
23  CREATE FUNCTION MONTHLY_TOTAL.SALES(MONTH INT)
24  RETURNS TABLE (TOTALMONTH.SALES DECIMAL(10,2))
25  LANGUAGE SQL
26  READS SQL DATA
27  NO EXTERNAL ACTION
28  DETERMINISTIC
29  RETURN SELECT (SELECT SUM(TOTAL_PRICE)
30     FROM SALES S, TIME T
31     WHERE S.TIME_ID = T.TIME_ID)
32  AS GRAND_TOTAL FROM
33  SYSIBM.SYSDUMMY1;
34
35  --- TO CALL THIS FUNCTION [db2 " SELECT * FROM TABLE(MONTHLY_TOTAL.SALES(1)) AS MANTHLY
36     "]
37  ----- BRANCH MANAGER -----
38  ---4)
39  CREATE VIEW TOP_BRANCH.SALES AS
40  SELECT * FROM (SELECT SALES.BR_ID, SUM(TOTAL_PRICE) AS TOTALEACH.BRANCH,
41     SUM(QTY) AS QTY_SOLD
42  FROM SALES
43  GROUP BY BR_ID
44  ORDER BY TOTALEACH.BRANCH DESC);
45
```



```

46
47
48 —5)
49 ———ALTER TABLE BRANCH ADD COLUMN BR_STATUS VARCHAR(15);
50
51 CREATE PROCEDURE ADD_STATUS
52 (IN ID CHAR(3) )
53 IF (SELECT SUM(QTY) FROM SALES WHERE SALES.BR_ID = ID)=0
54 THEN UPDATE BRANCH SET BR_STATUS = 'VERY BAD'
55 WHERE BRANCH.BR_ID = ID;
56
57 ELSEIF
58 (
59 SELECT SUM(QTY)
60 FROM SALES WHERE SALES.BR_ID = ID) BETWEEN 1 AND 10
61 THEN UPDATE BRANCH SET BR_STATUS = 'SATISFIED'
62 WHERE BRANCH.BR_ID = ID;
63
64 ELSEIF (SELECT SUM(QTY) FROM SALES WHERE SALES.BR_ID = ID) BETWEEN 21 AND 30
65 THEN UPDATE BRANCH SET BR_STATUS = 'GOOD'
66 WHERE BRANCH.BR_ID = ID;
67
68 ELSEIF (SELECT SUM(QTY) FROM SALES WHERE SALES.BR_ID = ID) BETWEEN 31 AND 60
69 THEN UPDATE BRANCH SET BR_STATUS = 'VERY GOOD'
70 WHERE BRANCH.BR_ID = ID;
71
72 ELSEIF (SELECT SUM(QTY) FROM SALES WHERE SALES.BR_ID = ID) > 60
73 THEN UPDATE BRANCH SET BR_STATUS = 'EXCELLENT '
74 WHERE BRANCH.BR_ID = ID;
75
76 END IF@
77
78 —6)
79 CREATE VIEW TOTAL_SOLD AS
80 SELECT (SELECT SUM(TOTAL.PRICE)
81 FROM SALES, BRANCH
82 WHERE BRANCH.BR_ID = SALES.BR_ID)
83 AS TOTAL_SOLD FROM
84 SYSIBM.SYSDUMMY1;
85
86 ————— CUSTOMER RELATIONSHIP MANAGER —————
87 —7)
88 CREATE VIEW BESTPRODUCTCUSTOMER AS
89 SELECT * FROM (SELECT CU_FNAME , SUM(QTY) AS TOTAL_ORDER_NUMBER
90 FROM SALES S, CUSTOMER C
91 WHERE S.CU_ID=C.CU_ID
92 GROUP BY CU_FNAME
93 ORDER BY TOTAL_ORDER_NUMBER DESC);
94

```

```

95
96 —8)
97
98 CREATE FUNCTION MONTHLY_TOTAL_SALES_BY_BRANCH(MONTH INT, BRANCH CHAR(3))
99 RETURNS TABLE (TOTAL_MONTH_SALES_BY_BRANCH DECIMAL(10,2))
100 LANGUAGE SQL
101 READS SQL DATA
102 NO EXTERNAL ACTION
103 DETERMINISTIC
104 RETURN SELECT (SELECT SUM(TOTAL_PRICE)
105   FROM SALES S, TIME T
106   WHERE S.TIME_ID = T.TIME_ID
107     AND S.BR_ID = BRANCH
108     AND T.MONTHNUMBER = MONTH)
109 AS GRAND_TOTAL FROM
110 SYSIBM.SYSDUMMY1;

```

6 Pivot tables and graphs based on Views

	A	B	C	D	E	F	G	H	I
1	Sum - TOTAL_INCOME	Data							
2	CAR_MODEL	1	2	3	4	6	7	8	Total Result
3	HONDA ACCORD			450000					450000
4	HONDA CITY				120000				120000
5	HONDA CIVIC	35000							35000
6	HONDA STREAM			210000					210000
7	MERCEDES CLS-CLASS		400000						400000
8	MERCEDES E-CLASS	300000							300000
9	MERCEDES SLK	500000							500000
10	PERODUA ALZA		60000						60000
11	PERODUA KANCIL						35000		35000
12	PERODUA MYVI							200000	200000
13	PROTON PERDANA		80000						80000
14	PROTON SAGA					48000			48000
15	PROTON SATRIA	20000							20000
16	PROTON WAJA			96000					96000
17	PROTON WIRA	12000							12000
18	TOYOTA 86		400000						400000
19	TOYOTA VELLFIRE	80000							80000
20	Total Result	947000	940000	756000	120000	48000	35000	200000	3046000

	A	B	C	D
1	Sum - TOTAL_EACH_BRANCH	Data		
2	BR_ID	8	10	Total Result
3	B01		370000	370000
4	B02		705000	705000
5	B03		774000	774000
6	B04		641000	641000
7	B05	556000		556000
8	Total Result	556000	2490000	3046000

	A	B
1	CU ENAN ▼	Sum - TOTAL_ORDER_NUMBER
2	Angela	1
3	Anna	1
4	Anne	1
5	Annie	1
6	Arthur	1
7	Beverly	1
8	Billy	1
9	Brandon	1
10	Christopher	1
11	Clarence	2
12	Diana	3
13	Diane	2
14	Donna	1
15	Doris	1
16	Elizabeth	1
17	Ernest	1
18	Henry	1
19	Jack	1
20	Janice	1
21	Joshua	1
22	Julia	1
23	Karen	2
24	Kathy	1
25	Kelly	1
26	Kenneth	1
27	Laura	1
28	Mark	1
29	Martin	1
30	Nicholas	1
31	Norma	1
32	Patrick	1
33	Peter	1
34	Philip	1
35	Rachel	1
36	Randy	1
37	Richard	1
38	Ruth	1
39	Sandra	1
40	Scott	1
41	Sharon	1
42	Shawn	1
43	Shirley	1
44	Timothy	1

7 Operational Database design and Data Warehouse

Operational database	Data warehouse
Stored with a functional or process orientation	Stored with a subject orientation
In operational database data are stored with a functional or process orientation. But when we move to the data warehouse we can identify that data are stored with subject orientation that facilitates multiple views for data and decision making.	

Operational database	Data warehouse
Different representation or meanings	Unified view of all data elements with a common definition and representation
Data warehouse concepts it unified view of all data elements with a common definition and representation. In operational database similar data are allowed different representations	

Operational database	Data warehouse
Represent current transactions	Historic in natural
Data are historic in nature in data warehouse. Here a dimension is added to facilitate data analysis and time corporations. But if we turn our eye to the operational database we can recognize that in operational database model data represent in current transactions. So there is a big difference in this perspective	

Operational database	Data warehouse
Update and delete	Add only periodically from operational system
In data warehouse data are changed. But here data only added periodical from operational systems. But when we turn our eyes to the operational database perspective we can recognize that data update and deleted are really common in operational database.	