MODUL - WEEK.09
DATA MANIPULATION (Sub Queries) SOME and ALL

I. DESKRIPSI TEMA

Construct query of SQL that suitable with the problem

II. CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

CLO5-SUB-CLO9:

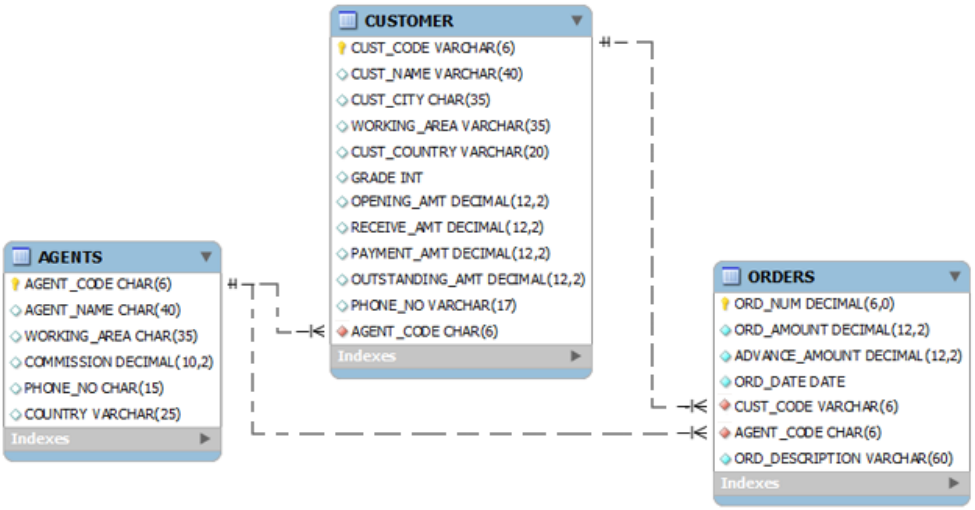
- Students are able to implement SQL Data Manipulation (C3)
- Students are able to create Sub Queries, Operator SOME and ALL(C6)

III. PENUNJANG PRAKTIKUM

1. Microsoft SQL Server management studio, SQL Server 2019
2. Module Practicum
3. These Module have been adapted from Connolly, T., & Begg, C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management. 6th edition. Pearson Education. USA. ISBN: 978-1-292-06118-4, Chapter 12&14

IV. LANGKAH-LANGKAH PRAKTIKUM

1. **Create database model:**



1.1 Here is the command to create the table agents:

```
CREATE TABLE  AGENTS
   (
    AGENT_CODE       CHAR(6) NOT NULL PRIMARY KEY,
    AGENT_NAME       CHAR(40),
    WORKING_AREA     CHAR(35),
    COMMISSION       DECIMAL(10,2),
    PHONE_NO         CHAR(15),
    COUNTRY          VARCHAR(25)
     );
```

1.2 Now insert records into the table agents:

AGENT_CODE	AGENT_NAME	WORKING_AREA	COMMISSION	PHONE_NO	COUNTRY
A001	Subbarao	Bangalore	0.14	077-12346674	
A002	Mukesh	Mumbai	0.11	029-12358964	
A003	Alex	London	0.13	075-12458969	
A004	Ivan	Torento	0.15	008-22544166	
A005	Anderson	Brisban	0.13	045-21447739	
A006	McDen	London	0.15	078-22255588	
A007	Ramasundar	Bangalore	0.15	077-25814763	
A008	Alford	New York	0.12	044-25874365	
A009	Benjamin	Hampshair	0.11	008-22536178	
A010	Santakumar	Chennai	0.14	007-22388644	
A011	Ravi Kumar	Bangalore	0.15	077-45625874	
A012	Lucida	San Jose	0.12	044-52981425	

1.3 Here is the command to create the table customer:

```
CREATE TABLE  CUSTOMER
(
    CUST_CODE    VARCHAR(6) NOT NULL PRIMARY KEY,
    CUST_NAME    VARCHAR(40) NOT NULL,
    CUST_CITY    CHAR(35),
    WORKING_AREA VARCHAR(35) NOT NULL,
    CUST_COUNTRY VARCHAR(20) NOT NULL,
    GRADE        DECIMAL,
    OPENING_AMT DECIMAL(12,2) NOT NULL,
    RECEIVE_AMT DECIMAL(12,2) NOT NULL,
    PAYMENT_AMT DECIMAL(12,2) NOT NULL,
    OUTSTANDING_AMT DECIMAL(12,2) NOT NULL,
    PHONE_NO VARCHAR(17) NOT NULL,
    AGENT_CODE CHAR(6) NOT NULL REFERENCES AGENTS
);
```

1.4 Now insert records into the table customer:

CUST_CODE	CUST_NAME	CUST_CITY	WORKING_AREA	CUST_COUNTRY	GRADE	OPENING_AMT	RECEIVE_AMT	PAYMENT_AMT	OUTSTANDING_AMT	PHONE_NO	AGENT_CODE
C00001	Micheal	New York	New York	USA	2	3000	5000	2000	6000	CCCCCCC	A008
C00002	Bolt	New York	New York	USA	3	5000	7000	9000	3000	DDNRDRH	A008
C00003	Martin	Torento	Torento	Canada	2	8000	7000	7000	8000	MJYURFD	A004
C00004	Winston	Brisban	Brisban	Australia	1	5000	8000	7000	6000	AAAAAAA	A005
C00005	Sasikant	Mumbai	Mumbai	India	1	7000	11000	7000	11000	147-25896312	A002
C00006	Shilton	Torento	Torento	Canada	1	10000	7000	6000	11000	DDDDDDD	A004
C00007	Ramanathan	Chennai	Chennai	India	1	7000	11000	9000	9000	GHRDWSD	A010
C00008	Karolina	Torento	Torento	Canada	1	7000	7000	9000	5000	HJKORED	A004
C00009	Ramesh	Mumbai	Mumbai	India	3	8000	7000	3000	12000	Phone No	A002
C00010	Charles	Hampshair	Hampshair	UK	3	6000	4000	5000	5000	MMMMMMM	A009
C00011	Sundariya	Chennai	Chennai	India	3	7000	11000	7000	11000	PPHGRTS	A010
C00012	Steven	San Jose	San Jose	USA	1	5000	7000	9000	3000	KRFYGJK	A012
C00013	Holmes	London	London	UK	2	6000	5000	7000	4000	BBBBBBB	A003
C00014	Rangarappa	Bangalore	Bangalore	India	2	8000	11000	7000	12000	AAAATGF	A001
C00015	Stuart	London	London	UK	1	6000	8000	3000	11000	GFSGERS	A003
C00016	Venkatpati	Bangalore	Bangalore	India	2	8000	11000	7000	12000	JRTVFDD	A007
C00017	Srinivas	Bangalore	Bangalore	India	2	8000	4000	3000	9000	AAAAAAB	A007
C00018	Fleming	Brisban	Brisban	Australia	2	7000	7000	9000	5000	NHBGVFC	A005
C00019	Yearannaidu	Chennai	Chennai	India	1	8000	7000	7000	8000	ZZZZBFV	A010
C00020	Albert	New York	New York	USA	3	5000	7000	6000	6000	BBBBSBB	A008
C00021	Jacks	Brisban	Brisban	Australia	1	7000	7000	7000	7000	WERTGDF	A005
C00022	Avinash	Mumbai	Mumbai	India	2	7000	11000	9000	9000	113-12345678	A002
C00023	Karl	London	London	UK	0	4000	6000	7000	3000	AAAABAA	A006
C00024	Cook	London	London	UK	2	4000	9000	7000	6000	FSDDSDF	A006

1.5 Here is the command to create the table orders:

```
CREATE TABLE  ORDERS
   (
    ORD_NUM      DECIMAL(6,0) NOT NULL PRIMARY KEY,
    ORD_AMOUNT   DECIMAL(12,2) NOT NULL,
    ADVANCE_AMOUNT  DECIMAL(12,2) NOT NULL,
    ORD_DATE     DATE NOT NULL,
    CUST_CODE    VARCHAR(6) NOT NULL REFERENCES CUSTOMER,
    AGENT_CODE   CHAR(6) NOT NULL REFERENCES AGENTS,
    ORD_DESCRIPTION VARCHAR(60) NOT NULL
   );
```

1.6 Now insert records into the table orders:

ORD_NUM	ORD_AMOUNT	ADVANCE_AMOUNT	ORD_DATE	CUST_CODE	AGENT_CODE	ORD_DESCRIPTION
200100	1000	600	8/1/2008	C00013	A003	SOD
200101	3000	1000	7/15/2008	C00001	A008	SOD
200102	2000	300	5/25/2008	C00012	A012	SOD
200103	1500	700	5/15/2008	C00021	A005	SOD
200104	1500	500	3/13/2008	C00006	A004	SOD
200105	2500	500	7/18/2008	C00025	A011	SOD
200106	2500	700	4/20/2008	C00005	A002	SOD
200107	4500	900	8/30/2008	C00007	A010	SOD
200108	4000	600	2/15/2008	C00008	A004	SOD
200109	3500	800	7/30/2008	C00011	A010	SOD
200110	3000	500	4/15/2008	C00019	A010	SOD
200111	1000	300	7/10/2008	C00020	A008	SOD
200112	2000	400	5/30/2008	C00016	A007	SOD
200113	4000	600	6/10/2008	C00022	A002	SOD
200114	3500	2000	8/15/2008	C00002	A008	SOD
200116	500	100	7/13/2008	C00010	A009	SOD
200117	800	200	10/20/2008	C00014	A001	SOD
200118	500	100	7/20/2008	C00023	A006	SOD
200119	4000	700	9/16/2008	C00007	A010	SOD
200120	500	100	7/20/2008	C00009	A002	SOD
200121	1500	600	9/23/2008	C00008	A004	SOD
200122	2500	400	9/16/2008	C00003	A004	SOD
200123	500	100	9/16/2008	C00022	A002	SOD
200124	500	100	6/20/2008	C00017	A007	SOD
200125	2000	600	10/10/2008	C00018	A005	SOD
200126	500	100	6/24/2008	C00022	A002	SOD
200127	2500	400	7/20/2008	C00015	A003	SOD
200128	3500	1500	7/20/2008	C00009	A002	SOD
200129	2500	500	7/20/2008	C00024	A006	SOD
200130	2500	400	7/30/2008	C00025	A011	SOD
200131	900	150	8/26/2008	C00012	A012	SOD
200133	1200	400	6/29/2008	C00009	A002	SOD
200134	4200	1800	9/25/2008	C00004	A005	SOD
200135	2000	800	9/16/2008	C00007	A010	SOD

2. SQL SOME Operator

Description:

- SOME compare a value to each value in a list or results from a query and evaluate to true if the result of an inner query contains at least one row.
- SOME must match at least one row in the subquery and must be preceded by comparison operators.
- Suppose using greater than (>) with SOME means greater than at least one value.
-

- **Syntax** :
  SELECT [column_name... | expression1]
  FROM [table_name]
  WHERE expression2 comparison_operator {ALL | ANY | SOME} (subquery)
- **Example**:
  To get 'agent_code', 'agent_name', 'working_area', 'commission' from 'agents' table with following conditions:
  'agent_code' should be within some 'agent_code' from 'customer' table, which satisfies the condition: 'cust_country' in the 'customer' table must be 'UK', the following SQL statement can be used :

```
SELECT agent_code,agent_name,working_area,commission
FROM agents
WHERE agent_code=SOME(
SELECT agent_code FROM customer
WHERE cust_country='UK');
```

- **Output**:

```
AGENT_CODE AGENT_NAME           WORKING_AREA         COMMISSION
---------- -------------------- -------------------- ----------
A009       Benjamin             Hampshair                   .11
A003       Alex                 London                      .13
A006       McDen                London                      .15
```

3. SQL ALL Operator

Description:

- ALL is used to select all records of a **SELECT STATEMENT**.
- It compares a value to every value in a list or results from a query.
- The ALL must be preceded by the comparison operators and evaluates to TRUE if the query returns no rows.
- For example, ALL means greater than every value, means greater than the maximum value. Suppose ALL (1, 2, 3) means greater than 3.
- **Syntax**:
  SELECT [column_name... | expression1]
  FROM [table_name]
  WHERE expression2 comparison_operator {ALL | ANY | SOME} (subquery)
- **Example**:
  To get 'des_date','des_amount' and 'ord_amount' columns from the 'despatch' table with following conditions 'des_amount' of 'despatch' table is more than 'ord_amount' from 'orders' table which satisfies the condition 'ord_amount' must be equal to 2000, the following SQL statement can be used:

```
SELECT des_date,des_amount,ord_amount
FROM despatch
WHERE des_amount>ALL(
SELECT ord_amount FROM orders
WHERE ord_amount=2000);
```

- **Output:**

```
DES_DATE  DES_AMOUNT ORD_AMOUNT
--------- ---------- ----------
12-JAN-08       3800       4000
19-OCT-08       4000       4000
24-JUL-08       4500       3500
```

REFERENSI

1. Connolly, T., & Begg, C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management. 6th edition. Pearson Education. USA. ISBN: 978-1-292-06118-4, Chapter 6