Name: Prathibha Pillai

Inventory management system Project

Database management system

What is container inventory management?

Container inventory and Management is an act of keeping track of each container a company (container lines) owns in various. It involves maintaining a list of number of containers at various location, their occupancy, their conditions etc. Container management system is a vital aspect of running a successful Container logistics operation facilitating best-in-class container utilization and lowest cost operations.

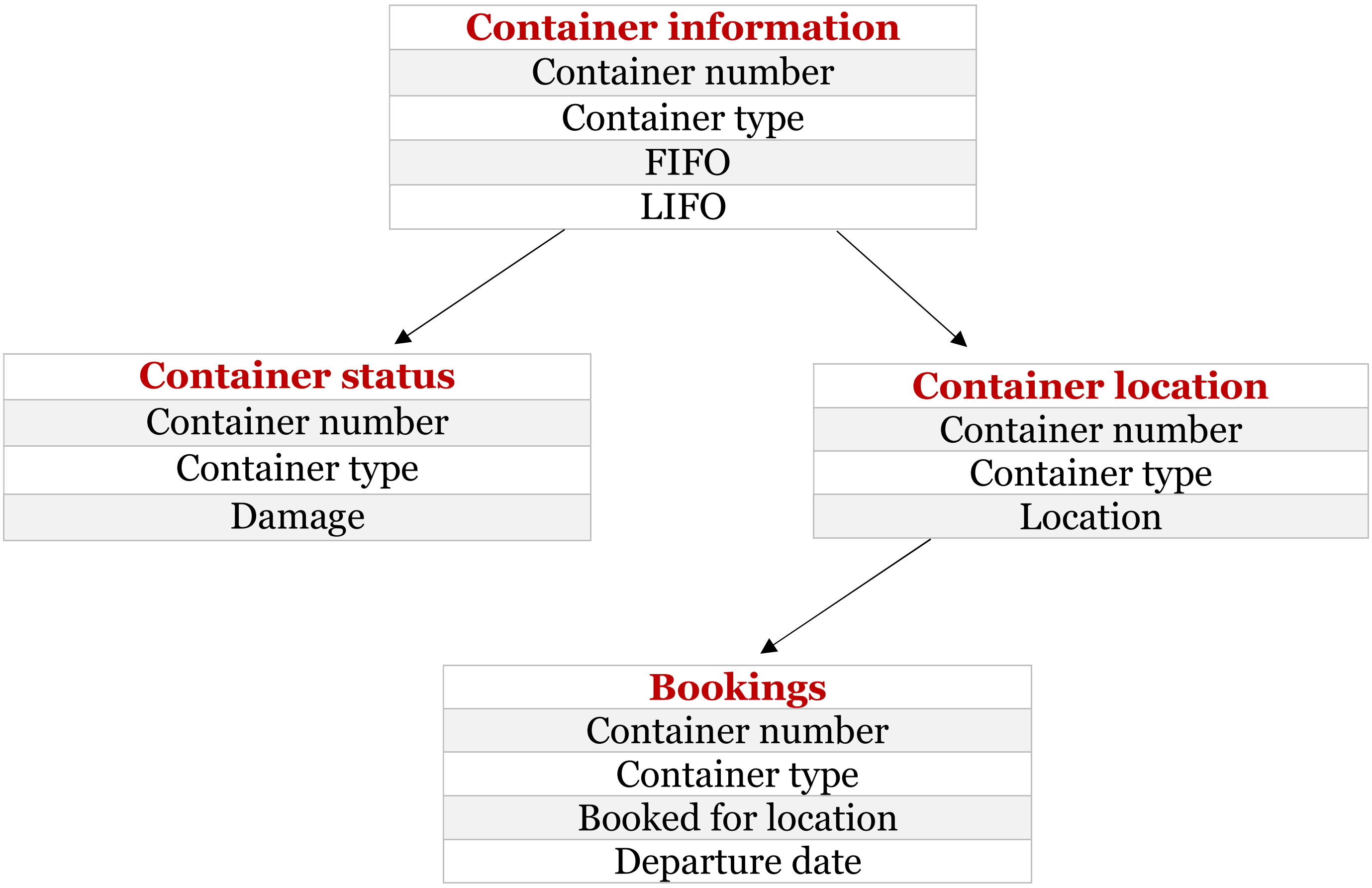
Container inventory is essential to make informed decisions and planning of the utilization of each container in the most cost-effective manner, as the constant utilization and movement of these containers is vital to maintain a profitable business.

So, Container Management System is not only just a tracking system; it is a full function solution that addresses all container inventory management, movements, and maintenance.

Objective of this project

- This project is about inventory management
- We can keep track of the whereabouts of the containers
- We can closely administer the utilization of each container and ensure maximum optimization is maintained
- We can keep track of maintenance requirement of each container
- We can see how many containers each location has, and in which locations the containers are available for utilization
- Data on which container are to follow FIFO or LIFO is also mentioned to make informed decisions

Overview of the database



Container information
Container number
Container type
FIFO
LIFO

Container status
Container number
Container type
Damage

Container location
Container number
Container type
Location

Bookings
Container number
Container type
Booked for location
Departure date

Structure of the tables

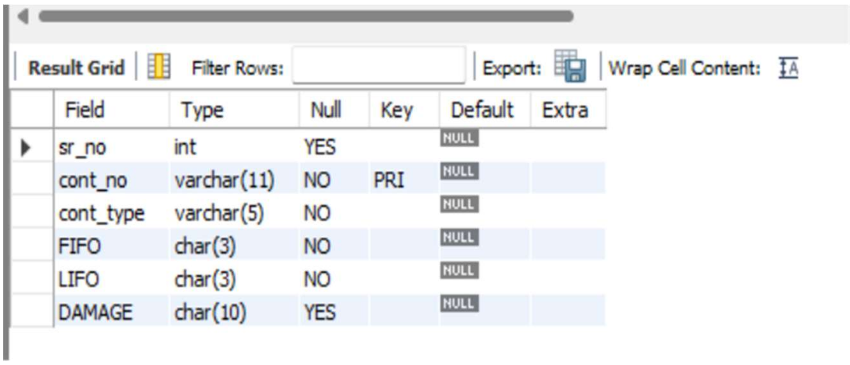
Container information (cont_info)

Syntax;

```
• create table cont_info (
    sr_no int,
    cont_no varchar (11) PRIMARY KEY,
    cont_type varchar(5) not null,
    FIFO char (3) not null,
    LIFO char (3) not null,
    DAMAGE char (10)
  );
```

```
• describe cont_info;
```

Output;



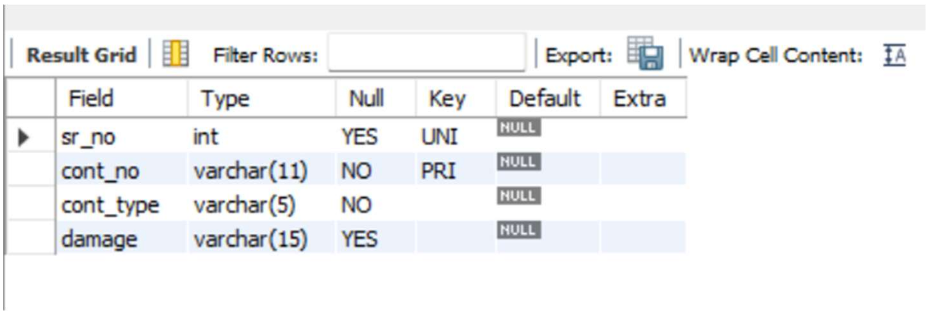
Field	Type	Null	Key	Default	Extra
sr_no	int	YES		NULL	
cont_no	varchar(11)	NO	PRI	NULL	
cont_type	varchar(5)	NO		NULL	
FIFO	char(3)	NO		NULL	
LIFO	char(3)	NO		NULL	
DAMAGE	char(10)	YES		NULL	

Container status (cont_status)

Syntax;

```
• create table cont_status(
  sr_no int unique key,
    cont_no varchar (11) PRIMARY KEY,
    cont_type varchar(5) not null,
    damage varchar(15)
  );
• describe cont_status;
```

Output;



Field	Type	Null	Key	Default	Extra
sr_no	int	YES	UNI	NULL	
cont_no	varchar(11)	NO	PRI	NULL	
cont_type	varchar(5)	NO		NULL	
damage	varchar(15)	YES		NULL	

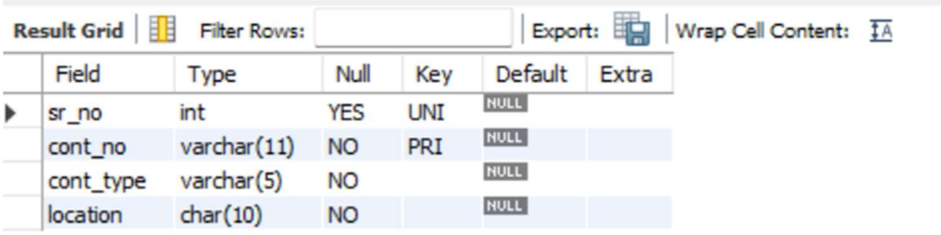
Location

Syntax;

-
  ```
  create table location(
  sr_no int unique key,
    cont_no varchar (11) PRIMARY KEY,
    cont_type varchar(5) not null,
    location char(10) not null
    );
  ```

-
  ```
  describe location;
  ```

Output;



Field	Type	Null	Key	Default	Extra
sr_no	int	YES	UNI	NULL	
cont_no	varchar(11)	NO	PRI	NULL	
cont_type	varchar(5)	NO		NULL	
location	char(10)	NO		NULL	

Bookings

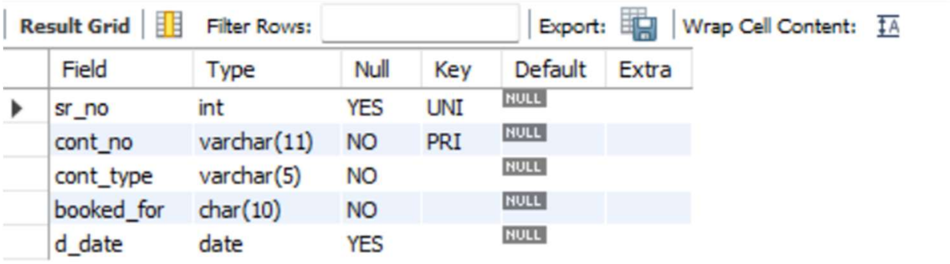
Syntax;

-
  ```
  create table bookings(
  sr_no int unique key,
    cont_no varchar (11) PRIMARY KEY,
    cont_type varchar(5) not null,
    booked_for char(10) not null,
    d_date int(10) date
  ```

);

- **describe** bookings;

Output;



Field	Type	Null	Key	Default	Extra
sr_no	int	YES	UNI	NULL	
cont_no	varchar(11)	NO	PRI	NULL	
cont_type	varchar(5)	NO		NULL	
booked_for	char(10)	NO		NULL	
d_date	date	YES		NULL	

Changes made to the structure and records of the tables

- The damage column in cont_info table is not required as we have a separate table to asses the container's condition. So, to drop the column the following query is used;
    - **alter table** cont_info **drop** damage;

- To add a foreign key after the table has been created
    - **alter table** bookings **add foreign key** (sr_no) **references** location (sr_no);
- To change the structure of a column;
    - **alter table** bookings **modify** booked_for char(10) **default null**;
    - **alter table** cont_status **alter** damage **set default** "no damage";
- To update the records in a table;
    - **update** cont_status **set** damage="under repair" **where** sr_no= 10;

- use of various **operators**;
    - **select** cont_no, sr_no, cont_type **from** cont_info **where** cont_type="40'DV";
    - **select** sr_no, cont_no, cont_type **from** location **where** cont_type="40'DV" **and** location= "Bahrain"; (and operator)
    - **select** sr_no, cont_no, cont_type **from** bookings **where** booked_for="Dammam" **or** booked_for="Jebel Ali"; (or operator)
    - **select** * **from** bookings **where** d_date **between** '2023-08-25' **and** '2023-08-30'; (range operator)

- To count records in a table;
    - **select** count(*) from cont_info;
    - **select** count(cont_type) **from** cont_info **group by** cont_type; (group by)

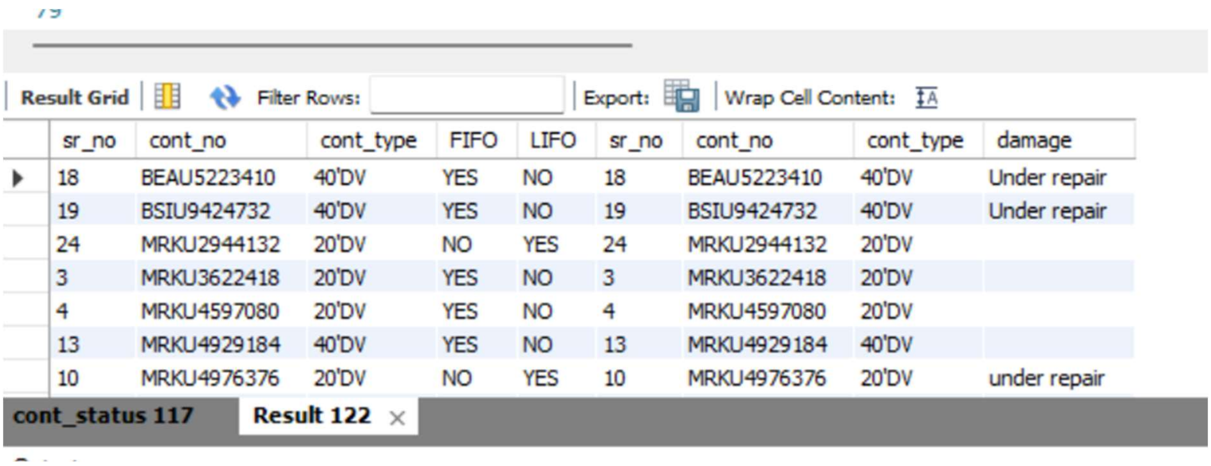
JOINS

- ✓ **Inner join** applied on tables cont_info and cont_status

Syntax;

select * **from** cont_info **join** cont_status **on** cont_info.cont_no= cont_status.cont_no;

Output;



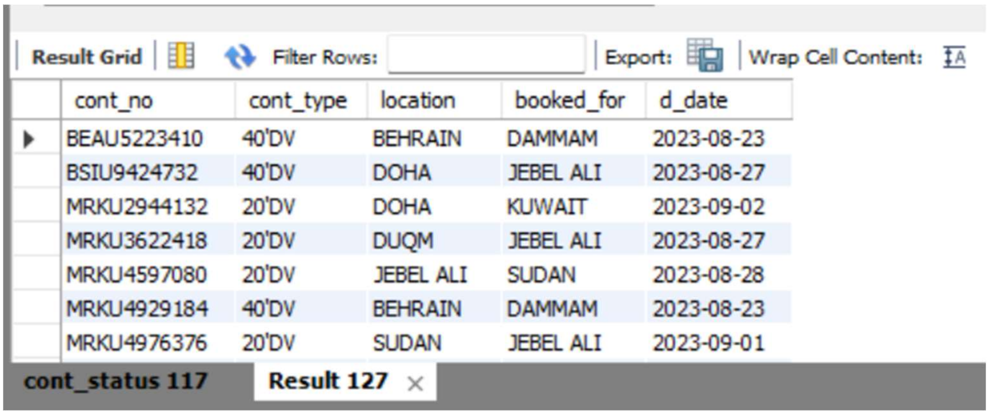
sr_no	cont_no	cont_type	FIFO	LIFO	sr_no	cont_no	cont_type	damage
18	BEAU5223410	40'DV	YES	NO	18	BEAU5223410	40'DV	Under repair
19	BSIU9424732	40'DV	YES	NO	19	BSIU9424732	40'DV	Under repair
24	MRKU2944132	20'DV	NO	YES	24	MRKU2944132	20'DV	
3	MRKU3622418	20'DV	YES	NO	3	MRKU3622418	20'DV	
4	MRKU4597080	20'DV	YES	NO	4	MRKU4597080	20'DV	
13	MRKU4929184	40'DV	YES	NO	13	MRKU4929184	40'DV	
10	MRKU4976376	20'DV	NO	YES	10	MRKU4976376	20'DV	under repair

- ✓ **Right join** applied on location and bookings tables to fetch the current location and for which location the containers are booked for with their departure dates

Syntax;

select lo.cont_no, lo.cont_type, lo.location, bo.booked_for, bo.d_date **from** location lo
right join bookings bo on lo.sr_no = bo.sr_no;

output;



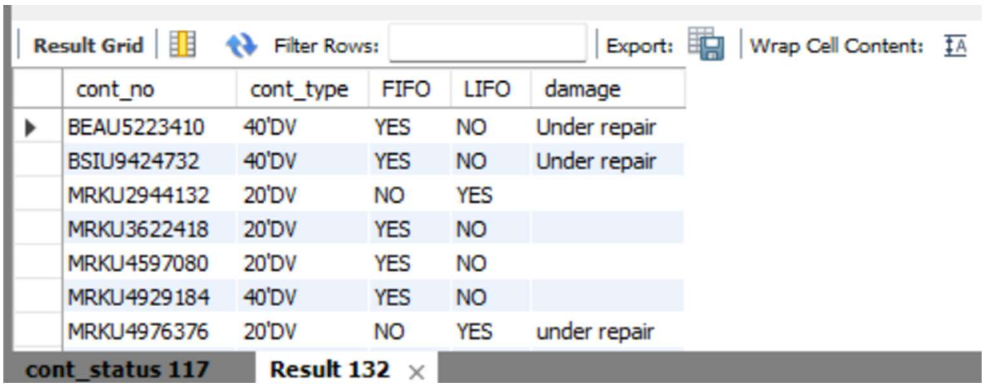
cont_no	cont_type	location	booked_for	d_date
BEAU5223410	40'DV	BEHRAIN	DAMMAM	2023-08-23
BSIU9424732	40'DV	DOHA	JEBEL ALI	2023-08-27
MRKU2944132	20'DV	DOHA	KUWAIT	2023-09-02
MRKU3622418	20'DV	DUQM	JEBEL ALI	2023-08-27
MRKU4597080	20'DV	JEBEL ALI	SUDAN	2023-08-28
MRKU4929184	40'DV	BEHRAIN	DAMMAM	2023-08-23
MRKU4976376	20'DV	SUDAN	JEBEL ALI	2023-09-01

- ✓ **Left join** applied on cont_info and cont_status to fetch details on damaged containers and which containers come under FIFO or LIFO inventory valuation method

  Syntax;

  select ci.cont_no, ci.cont_type, ci.FIFO, ci.LIFO, cs.damage **from** cont_info ci **left join** cont_status cs **on** ci.sr_no = cs.sr_no;

  output;



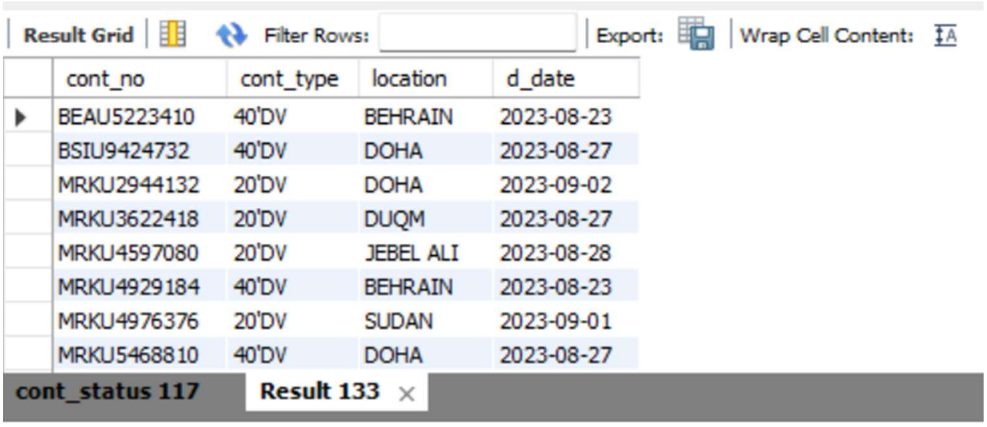
cont_no	cont_type	FIFO	LIFO	damage
BEAU5223410	40'DV	YES	NO	Under repair
BSIU9424732	40'DV	YES	NO	Under repair
MRKU2944132	20'DV	NO	YES	
MRKU3622418	20'DV	YES	NO	
MRKU4597080	20'DV	YES	NO	
MRKU4929184	40'DV	YES	NO	
MRKU4976376	20'DV	NO	YES	under repair

- ✓ **Left outer join** applied on location and bookings to fetch the date at which each container would depart from its respective current location

  Syntax;

  select location.cont_no, location.cont_type, location.location, bookings.d_date **from** location **left outer join** bookings **on** location.sr_no=bookings.sr_no **order by** location.cont_no;

  output;



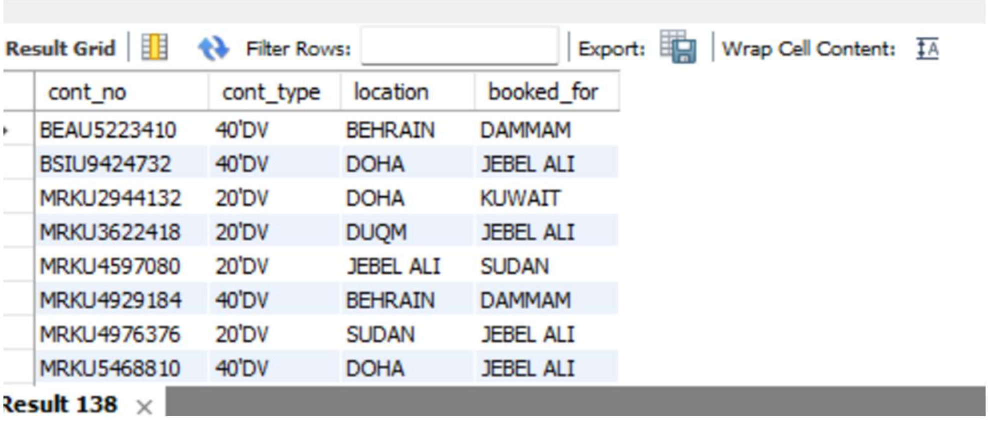
cont_no	cont_type	location	d_date
BEAU5223410	40'DV	BEHRAIN	2023-08-23
BSIU9424732	40'DV	DOHA	2023-08-27
MRKU2944132	20'DV	DOHA	2023-09-02
MRKU3622418	20'DV	DUQM	2023-08-27
MRKU4597080	20'DV	JEBEL ALI	2023-08-28
MRKU4929184	40'DV	BEHRAIN	2023-08-23
MRKU4976376	20'DV	SUDAN	2023-09-01
MRKU5468810	40'DV	DOHA	2023-08-27

- ✓ **Right outer join** is used to fetch the current location and location for with the container is booked for

  Syntax;

  select location.cont_no, location.cont_type, location.location, bookings.booked_for
  from location **right outer join** bookings **on** location.sr_no=bookings.sr_no
  order by location.cont_no;

  output;



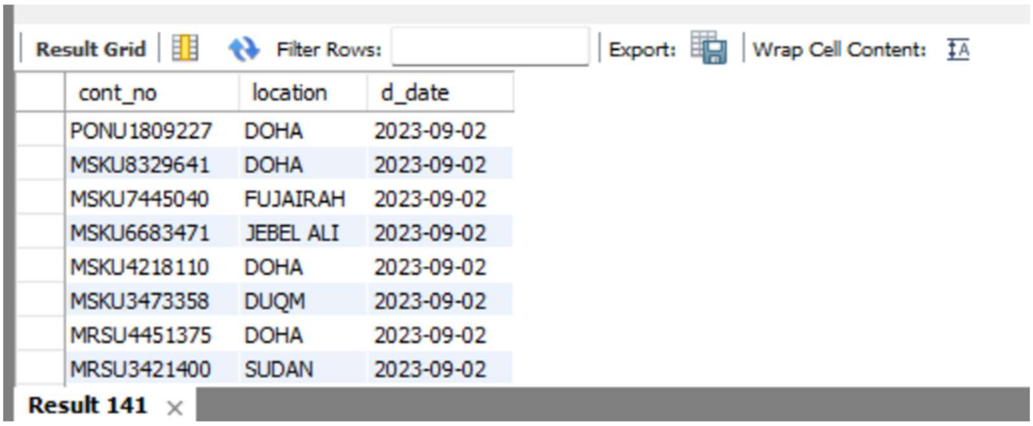
cont_no	cont_type	location	booked_for
BEAU5223410	40'DV	BEHRAIN	DAMMAM
BSIU9424732	40'DV	DOHA	JEBEL ALI
MRKU2944132	20'DV	DOHA	KUWAIT
MRKU3622418	20'DV	DUQM	JEBEL ALI
MRKU4597080	20'DV	JEBEL ALI	SUDAN
MRKU4929184	40'DV	BEHRAIN	DAMMAM
MRKU4976376	20'DV	SUDAN	JEBEL ALI
MRKU5468810	40'DV	DOHA	JEBEL ALI

- ✓ **Cross join** is used on location and bookings to fetch all the records in location and departure date from respective tables

  Syntax;

  select location.cont_no, location.location, bookings.d_date **from** location **cross join** bookings;

  output;



cont_no	location	d_date
PONU1809227	DOHA	2023-09-02
MSKU8329641	DOHA	2023-09-02
MSKU7445040	FUJAIRAH	2023-09-02
MSKU6683471	JEBEL ALI	2023-09-02
MSKU4218110	DOHA	2023-09-02
MSKU3473358	DUQM	2023-09-02
MRSU4451375	DOHA	2023-09-02
MRSU3421400	SUDAN	2023-09-02

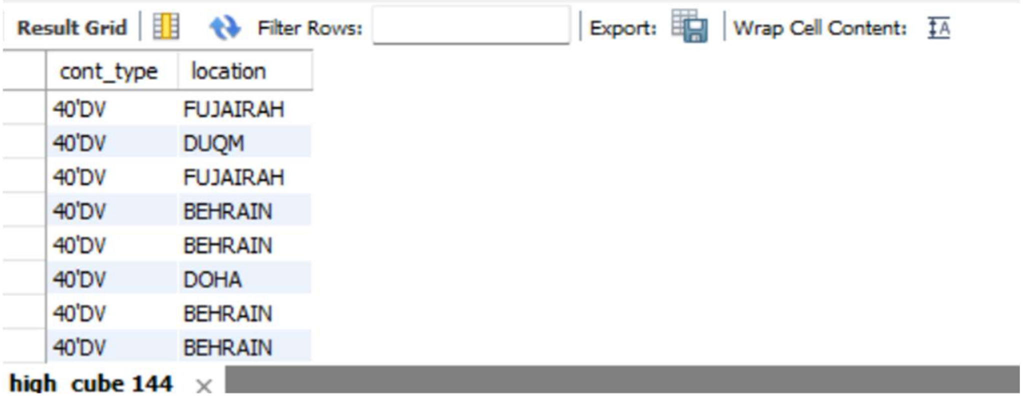
Views

- Views table high cube created to display all the 40 high cube containers in circulation

  Syntax;

  create view high_cube **as select** cont_type, location **from** location **where** cont_type="40'dv";
  select * **from** high_cube;

output;

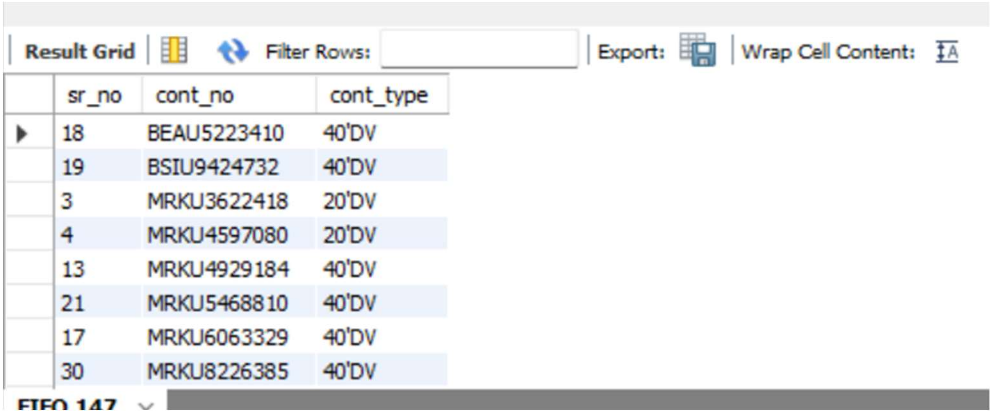


cont_type	location
40'DV	FUJAIRAH
40'DV	DUQM
40'DV	FUJAIRAH
40'DV	BEHRAIN
40'DV	BEHRAIN
40'DV	DOHA
40'DV	BEHRAIN
40'DV	BEHRAIN

- Views table FIFO created to display all the containers handled in FIFO method

  Syntax;
  create view FIFO **as select** sr_no, cont_no, cont_type **from** cont_info **where** FIFO='yes';
  select * **from** FIFO;

output;



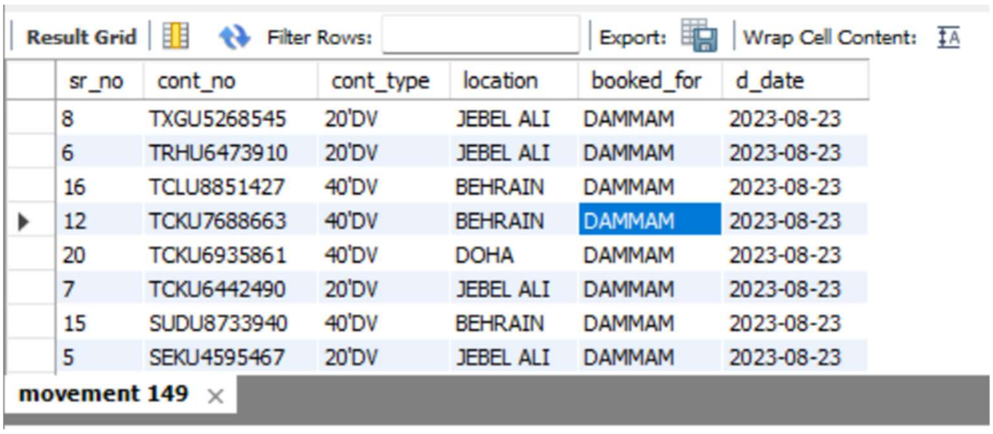
sr_no	cont_no	cont_type
18	BEAU5223410	40'DV
19	BSIU9424732	40'DV
3	MRKU3622418	20'DV
4	MRKU4597080	20'DV
13	MRKU4929184	40'DV
21	MRKU5468810	40'DV
17	MRKU6063329	40'DV
30	MRKU8226385	40'DV

- View table movement created with tables location and bookings to understand the movement of the containers

Syntax;

create view movement **as select** location.sr_no, location.cont_no, location.cont_type, location.location, bookings.booked_for, d_date **from** location, bookings;
select * **from** movement;

output;



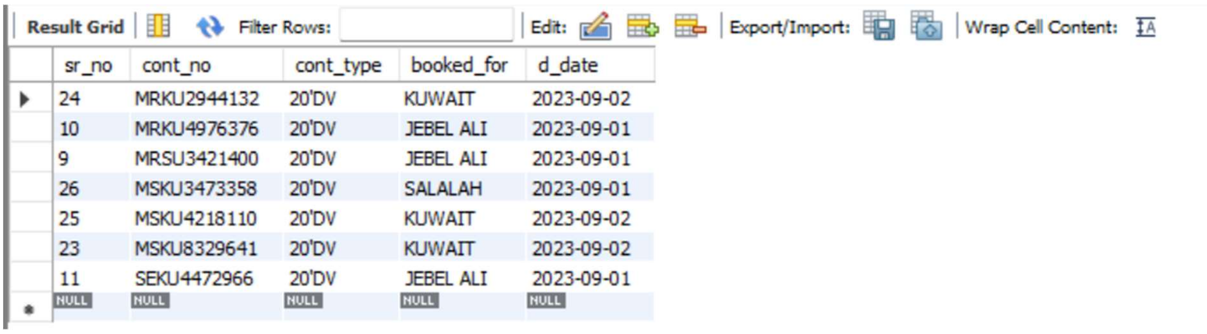
sr_no	cont_no	cont_type	location	booked_for	d_date
8	TXGU5268545	20'DV	JEBEL ALI	DAMMAM	2023-08-23
6	TRHU6473910	20'DV	JEBEL ALI	DAMMAM	2023-08-23
16	TCLU8851427	40'DV	BEHRAIN	DAMMAM	2023-08-23
12	TCKU7688663	40'DV	BEHRAIN	DAMMAM	2023-08-23
20	TCKU6935861	40'DV	DOHA	DAMMAM	2023-08-23
7	TCKU6442490	20'DV	JEBEL ALI	DAMMAM	2023-08-23
15	SUDU8733940	40'DV	BEHRAIN	DAMMAM	2023-08-23
5	SEKU4595467	20'DV	JEBEL ALI	DAMMAM	2023-08-23

Sub-query

- Sub-query to fetch 20’DV containers departing between 29-08-2023 and 02-09-2023

Syntax;
select * **from** bookings **where** sr_no **in**(select sr_no **from** location where cont_type="20'DV") **having** d_date **between** '2023-08-29' **and** '2023-09-02';

output;

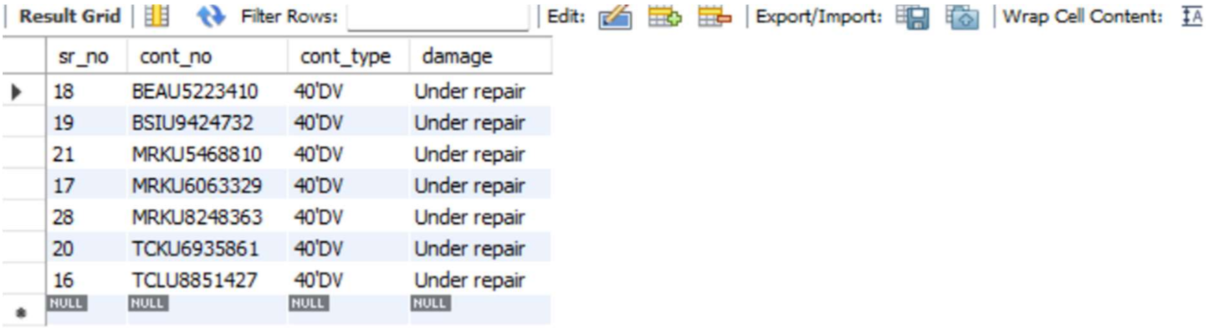


sr_no	cont_no	cont_type	booked_for	d_date
24	MRKU2944132	20'DV	KUWAIT	2023-09-02
10	MRKU4976376	20'DV	JEBEL ALI	2023-09-01
9	MRSU3421400	20'DV	JEBEL ALI	2023-09-01
26	MSKU3473358	20'DV	SALALAH	2023-09-01
25	MSKU4218110	20'DV	KUWAIT	2023-09-02
23	MSKU8329641	20'DV	KUWAIT	2023-09-02
11	SEKU4472966	20'DV	JEBEL ALI	2023-09-01
NULL	NULL	NULL	NULL	NULL

- Sub-query to fetch 40’dv containers that are under repair from tables cont_info and cont_status

Syntax;
select * **from** cont_status where sr_no **in**(select sr_no from cont_info **where** cont_type="40'DV") **having** damage='under repair';

output;



sr_no	cont_no	cont_type	damage
18	BEAU5223410	40'DV	Under repair
19	BSIU9424732	40'DV	Under repair
21	MRKU5468810	40'DV	Under repair
17	MRKU6063329	40'DV	Under repair
28	MRKU8248363	40'DV	Under repair
20	TCKU6935861	40'DV	Under repair
16	TCLU8851427	40'DV	Under repair
NULL	NULL	NULL	NULL

Thank You