

CARGO AGENCY

DATABSEEEXAMPLE



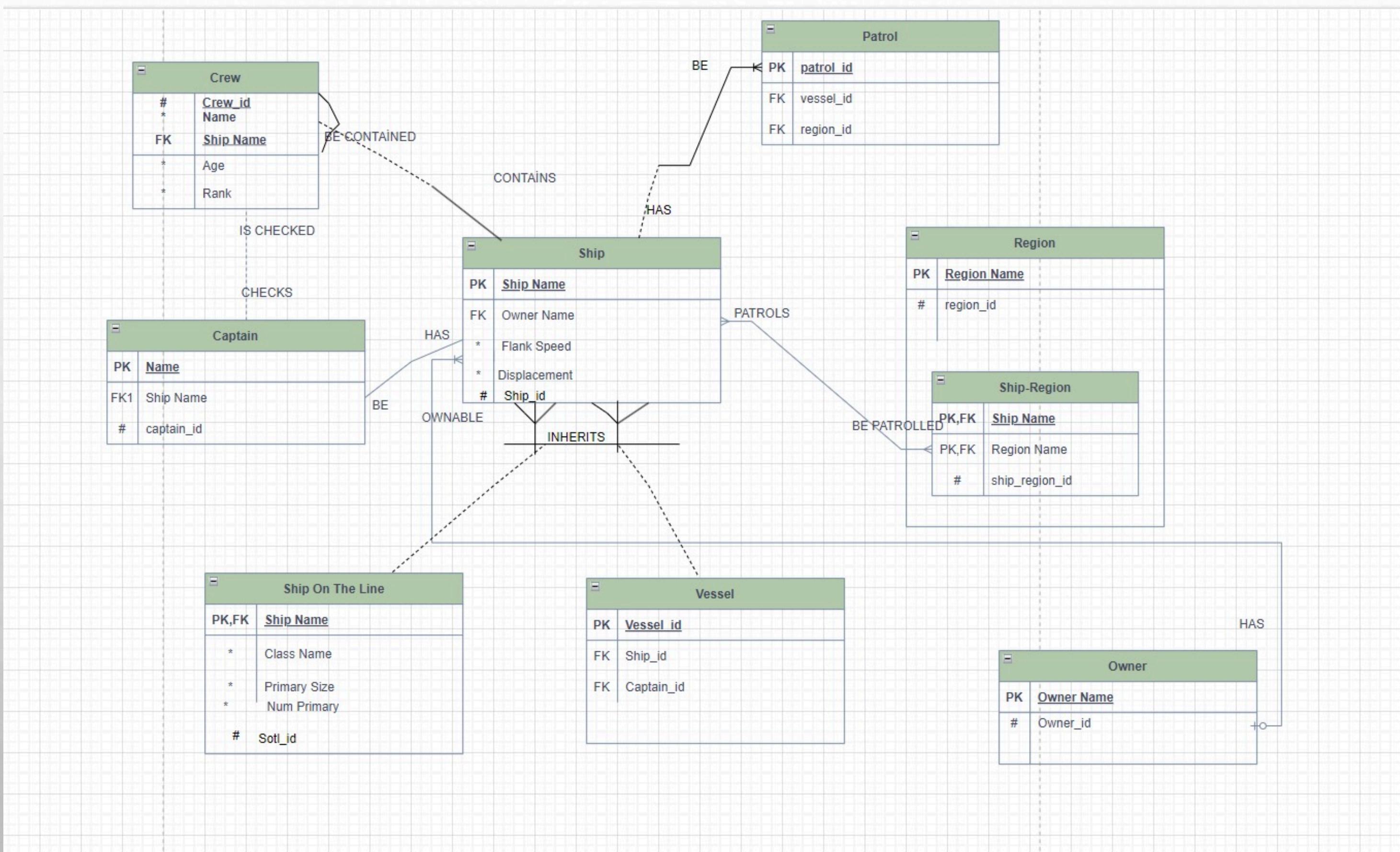
By Group 
B orc elle

Scenario:

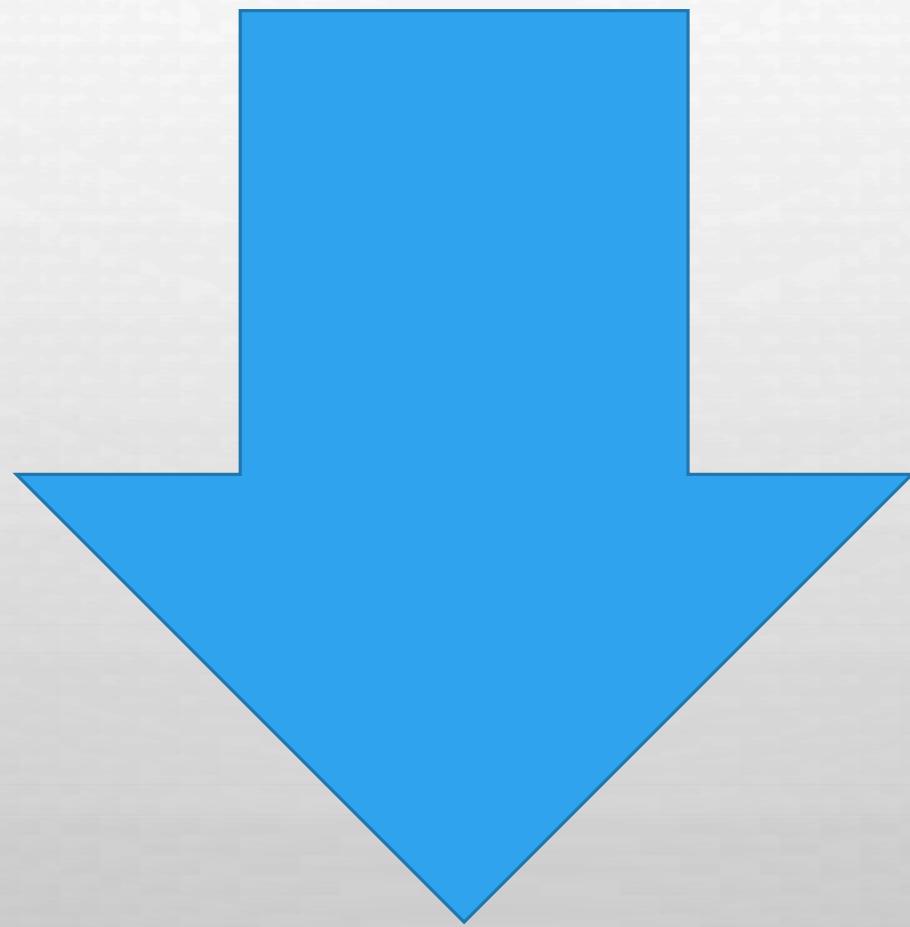
Cargo Management We aim to develop a cargo management system where a crew of people is controlled by a captain. Each crew can only be assigned to one vessel and is under the command of the captain in charge of that vessel. Ships are classified into two types: Vessel and SOTL (Ship of the Line). These ships patrol various regions to ensure security. Multiple ships can patrol multiple regions simultaneously. Crew members are uniquely identified by their names and also have information such as age and rank. Captains are also uniquely identified by their names, and each captain commands one or more crew members. Each ship is uniquely identified by its name and includes details such as flank speed and displacement. SOTL ships have additional attributes like class name, primary size, and the number of primary weapons. Vessel ships include information about the type of vessel. Regions are uniquely identified by their names and can be patrolled by multiple ships. Additionally, each ship has an owner, which indicates the ownership of the ships. This structure allows the cargo management system to effectively manage the relationships between crew members, captains, ships, regions, and ship types.

ERD

- 1. ship is super type**
- 2. Sotl and vessel are sub-types of ship**
- 3. there is hierarchical relationship between ship, vessel and sotl**



MatrixDiagram Defining all necessary relationships



ENTITIES

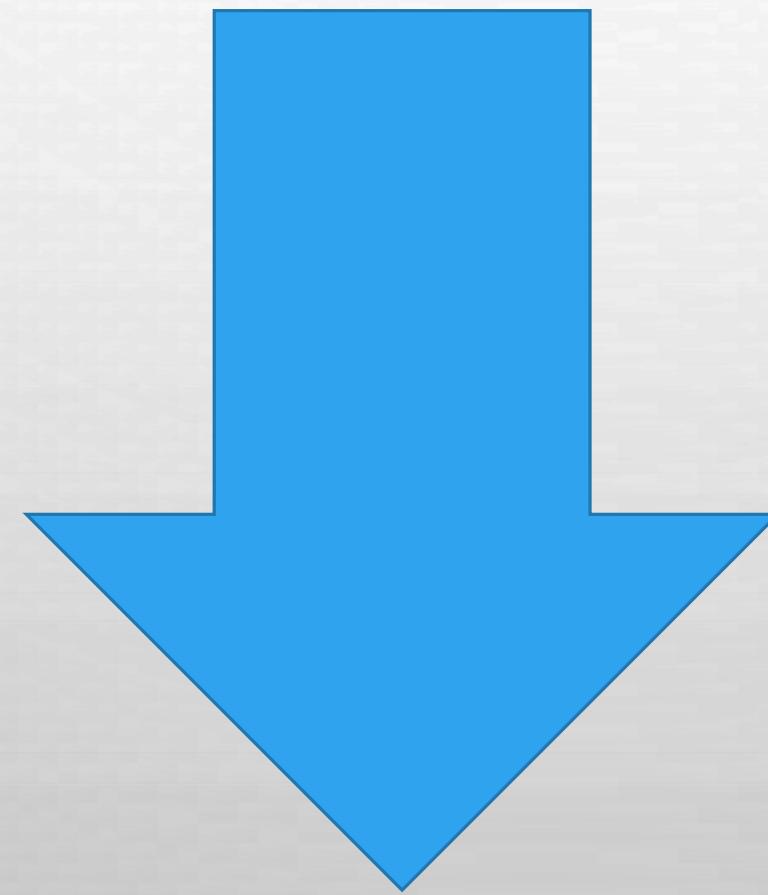
Entity-1	Entity-2	Entity-3	Entity-4	Entity-5	Entity-6	Entity-7	Entity-8	Entity-9
Crew	Captain	Ship	patrol	Region	Ship-Region	Ship On The Line	Vessels	Owner

Attributes:

Attributes	Entity-1	Entity-2	Entity-3	Entity-4	Entity-5	Entity-6	Entity-7	Entity-8	Entity-9
	Crew	Captain	Ship	Patrol	Region	Ship-Region	Ship On The Line	Vessel	Owner
Attributes-1	Name	Name	ShipName	patrol_id (PK)	RegionName	ShipName	ShipName	Vessel_id	OwnerName
Attributes-2	Age	ShipName	FlankSpeed			RegionName	ClassName		
Attributes-3	Rank		Displacement	region_id (FK)			PrimarySize		
Attributes-4	ShipName		OwnerName	vessel_id (FK)			NumPrimary	Ship_id (FK)	

Relationships Matrix Diagram

- Table instance charts map ERD to DB structure



CREW (CRW)		
Key Type	Optionalit	Column Name
pk	y * * * *	crew_id
		name
		rank
		assigned_ship

SHIP (SHP)		
Key Type	Optionality	ColumnName
pk	*	ship_id
fk	*	captain_id
	*	name
	*	type
	*	capacity

CAPTAIN (CPT)		
Key Type	Optionality	Column Name
pk	*	captain_id
fk	*	crew_id

PATROL (PTL)		
Key Type	Optionality	ColumnName
pk	*	patrol_id
fk	*	ship_id
	*	date

REGION (RGN)		
Key Type	Optionalit	ColumnName
pk	y * *	region_id
		name

SHIP-REGION (SHP-RGN)		
Key Type	Optionality	Column Name
pk	*	ship_region_id
fk	*	ship_id
fk	*	region_id

VESSEL (VSL)		
Key Type	Optionalit	Column Name
pk	y *	vessel_id
fk	*	ship_id

SHIP ON THE LINE (SOTL)		
Key Type	Optionality	Column Name
Pk	*	ship_on_the_line_id
fk	*	ship_id

OWNER (OWN)		
Key Type	Optionality	Column Name
pk	*	
fk	*	owner_id
fk	*	name

address

CREATE DATABASE

```
CREATE TABLE Crew (
    name VARCHAR(255) PRIMARY KEY,
    ship_name VARCHAR(255),
    age INT,
    rank VARCHAR(255),
    FOREIGN KEY (ship_name) REFERENCES Ship(ship_name)
);
```

```
CREATE TABLE Ship (
    ship_nameVARCHAR(255) PRIMARY KEY,
    owner_name VARCHAR(255),
    flank_speedDECIMAL(5, 2),
    displacement DECIMAL(10, 2),
    FOREIGN KEY (owner_name) REFERENCES Owner(owner_name)
);
```

```
CREATE TABLE Ship (
    ship_nameVARCHAR(255) PRIMARY KEY,
    owner_name VARCHAR(255),
    flank_speedDECIMAL(5, 2),
    displacement DECIMAL(10, 2),
    FOREIGN KEY (owner_name) REFERENCES Owner(owner_name)
);
```

```
CREATE TABLE Captain (
    name VARCHAR(255) PRIMARY KEY,
    ship_name VARCHAR(255),
    FOREIGN KEY (ship_name) REFERENCES Ship(ship_name)
);
```

```
CREATE TABLE Ship_On_The_Line(
    ship_name VARCHAR(255),
    class_name VARCHAR(255),
    num_primary INT,
    primary_sizeDECIMAL(10, 2),
    PRIMARY KEY (ship_name),
    FOREIGN KEY (ship_name) REFERENCES Ship(ship_name)
);
```

```
CREATE TABLE Vessel (
    vessel_idINT PRIMARY KEY,
    ship_id VARCHAR(255),
    captain_id VARCHAR(255),
    FOREIGN KEY (ship_id) REFERENCES Ship(ship_name),
    FOREIGN KEY (captain_id) REFERENCES Captain(name)
);
```

```
CREATE TABLE Patrol (
    patrol_id INT PRIMARY KEY,
    vessel_id INT,
    region_id VARCHAR(255),
    FOREIGN KEY (vessel_id) REFERENCES Vessel(vessel_id),
    FOREIGN KEY (region_id) REFERENCES Region(region_name)
);
```

```
CREATE TABLE Region (
    region_name VARCHAR(255) PRIMARY KEY
);
```

```
CREATE TABLE Ship_Region(
    ship_name VARCHAR(255),
    region_name VARCHAR(255),
    PRIMARY KEY (ship_name, region_name),
    FOREIGN KEY (ship_name) REFERENCES Ship(ship_name),
    FOREIGN KEY (region_name) REFERENCES Region(region_name)
);
```

```
CREATE TABLE Owner (
    owner_name VARCHAR(255) PRIMARY KEY
);
```

Inserting Data

```
INSERT INTO SHIP (ship_id, ship_name, region_id) VALUES  
(1, 'Intrepid', 1), (2, 'Enterprise', 2), (3,  
'Voyager', 3), (4, 'Defiant', 1), (5, 'Titan', 2), (6,  
'Discovery', 3);
```

```
'INSERT INTO REGION (region_id, region_name) VALUES  
(1, 'Intrepid'),  
(2, 'Atlantic Ocean'),  
(3, 'Pacific Ocean');
```

```
INSERT INTO CREW (crew_id, crew_name, age, rank, crew_id) VALUES  
(1, 'John Doe', 30, 'Lieutenant Commander', 1),  
(2, 'Jane Smith', 25, 'Ensign', 2),  
(3, 'Jim Brown', 35, 'Lieutenant', 3),  
(4, 'Lisa White', 28, 'Lieutenant', 1),  
(5, 'Tom Black', 40, 'Commander', 2),  
(6, 'Emily Green', 23, 'Ensign', 3);
```

```
ALTER TABLE CREW
ADD REGION VARCHAR(100);
COLUMN
UPDATE CREW
SET REGION = CASE
WHEN RANK = 'LIEUTENANT COMMANDER'
    THEN 'REGION 1'
WHEN RANK = 'ENSIGN' THEN 'REGION 2'
WHEN RANK = 'LIEUTENANT' THEN 'REGION 3'
    ELSE 'UNKNOWN'
END;
```

```
UPDATE CREW
SET REGION = 'NEW REGION'
WHERE RANK = 'LIEUTENANT' AND REGION =
    'REGION 3';
```

QUERIES

1_one statement including subquery:

Query to Select Crew Members Younger than the Average Age

```
SELECT crew_name, age, rank
```

```
FROM CREW
```

```
WHERE age < (SELECT AVG(age) FROM CREW);
```



A screenshot of a SQL command-line interface showing the execution of the provided query. The results table displays three crew members: Carol Taylor, Dave Brown, and Alice Johnson, all of whom are younger than the average age of 30.

CREW_NAME	AGE	RANK
Carol Taylor	30	Lieutenant Commander
Dave Brown	26	Ensign
Alice Johnson	29	Ensign

3 rows returned in 0.00 seconds Download

QUERIES

2_SQL statement including a join

Query to Retrieve Ship Names in the 'Intrepid' Region

- SELECT S.SHIP_NAME
- FROM SHIP S
- INNER JOIN REGION R ON S.REGION_ID= R.REGION_ID
- WHERE R.REGION_NAME= 'INTREPID';



```
SQL Workshop - Team Development - Gallery
SQL Commands
Language: SQL Rows: 10
Clear Command Find Tables
1 SELECT s.ship_name
2 FROM SHIP s
3 INNER JOIN REGION r ON s.ship_id = r.region_id
4 WHERE r.region_name = 'Atlantic Ocean';
5
6
7
8
9
10
11
12
13
14
15

Results Explain Describe Saved SQL History
SHIP_NAME
SS Enterprise
1 rows returned in 0.00 seconds Download
```

QUERIES

3_SQLstatementincluding agroupby

Query to Count Total Crew Members by Region

```
SELECT r.region_name AS region_name, COUNT(*) AS total_crew  
FROM Crew c  
INNER JOIN Region r ON c.CREW_ID = r.REGION_ID  
GROUP BY r.region_name;
```



Screenshot of a SQL query editor showing the results of the provided query.

The query:

```
1 SELECT r.region_name AS region_name, COUNT(*) AS total_crew  
2 FROM Crew c  
3 INNER JOIN Region r ON c.CREW_ID = r.REGION_ID  
4 GROUP BY r.region_name;
```

The results table:

REGION_NAME	TOTAL_CREW
Arctic Ocean	1
Atlantic Ocean	1
Indian Ocean	1
Mediterranean Sea	1

QUERIES

4_SQL statement including a date function

Query to Calculate and Display Age of Crew Members

```
SELECT crew_name, birth_date, DATE_PART('year', AGE(birth_date)) AS age  
FROM Crew;
```



Screenshot of a SQL query editor showing the execution of the provided SQL code and its results.

The SQL code is:

```
1 SELECT crew_name, birth_date,  
2       TRUNC(MONTHS_BETWEEN(SYSDATE, birth_date) / 12) AS age,  
3       RANK  
4 FROM Crew  
5 WHERE TRUNC(MONTHS_BETWEEN(SYSDATE, birth_date) / 12) < 30;
```

The results table shows the following data:

CREW_NAME	BIRTH_DATE	AGE	RANK
David Brown	05-Dec-1998	25	Ensign
Alice Johnson	20-Aug-1995	28	Ensign

2 rows returned in 0.00 seconds

QUERIES

5_ SQL statement including a character function

Query to Convert Crew Member Names to Uppercase

```
SELECT UPPER(crew_name) AS  
crew_name_upper
```

```
FROM Crew;
```



The screenshot shows a SQL command window with the following details:

- SQL Commands tab:** The tab is selected, showing the SQL code.
- Language dropdown:** Set to "SQL".
- Rows dropdown:** Set to "10".
- Buttons:** Includes "Clear Command" and "Find Tables".
- SQL Query:**

```
1 SELECT UPPER(crew_name) AS crew_name_upper  
2 FROM Crew;
```
- Results tab:** The tab is also selected, showing the results of the query.
- Result Headers:** The column is labeled "CREW_NAME_UPPER".
- Result Data:** The names listed are ALICE JOHNSON, BOB WILLIAMS, CAROL TAYLOR, and DAVE BROWN.

QUERIES

6_SQL statement including an update Update Crew Regions Based on Rank

SET region = CASE

WHEN RANK = 'Lieutenant Commander' THEN 'Region 1'

WHEN RANK = 'Ensign' THEN 'Region 2'

WHEN RANK = 'Lieutenant' THEN 'Region 3'

--Add other cases for other ranks if needed

ELSE 'Unknown'

END;//ranka göre region atadık



Language SQL Rows 10 Clear Command Find Tables

```
1 UPDATE Crew
2 SET region = CASE
3 WHEN RANK = 'Lieutenant Commander' THEN 'Region 1'
4 WHEN RANK = 'Ensign' THEN 'Region 2'
5 WHEN RANK = 'Lieutenant' THEN 'Region 3'
6 -- Add other cases for other ranks if needed
7 ELSE 'Unknown'
8 END;
```

Results Explain Describe Saved SQL History

5 row(s) updated.

0.00 seconds

QUERIES

7_SQL statement including an alter table:

Add Region Column to Crew Table

ALTER TABLE Crew

ADD region VARCHAR(100);



```
SQL Commands
Language SQL Rows 10
ALTERTABLE Crew
ADDregion VARCHAR(100)
Table altered.
0.02 seconds
```