Database Report - Council Lifeguards

Section A

Application Description

The database I designed was for lifeguards working for a county council. There are 7 relational tables in the design. Beaches, qualifications, lifeguards, managers, and equipment are all things that are used in my job as a beach lifeguard. The database is based on Fingal County Council beaches as there are 13 of these and varying amounts of equipment.

The beaches table consists of the beach name, the beach id and the number of lifeguard stations. Larger beaches would have more stations and more lifeguards as well as more equipment.

Qualifications contain qualifications a lifeguard can get with a qualification id, the type, the company it was received from and the duration it lasts for in years.

The managers table contains the name of the manager, their hourly pay and their pay number.

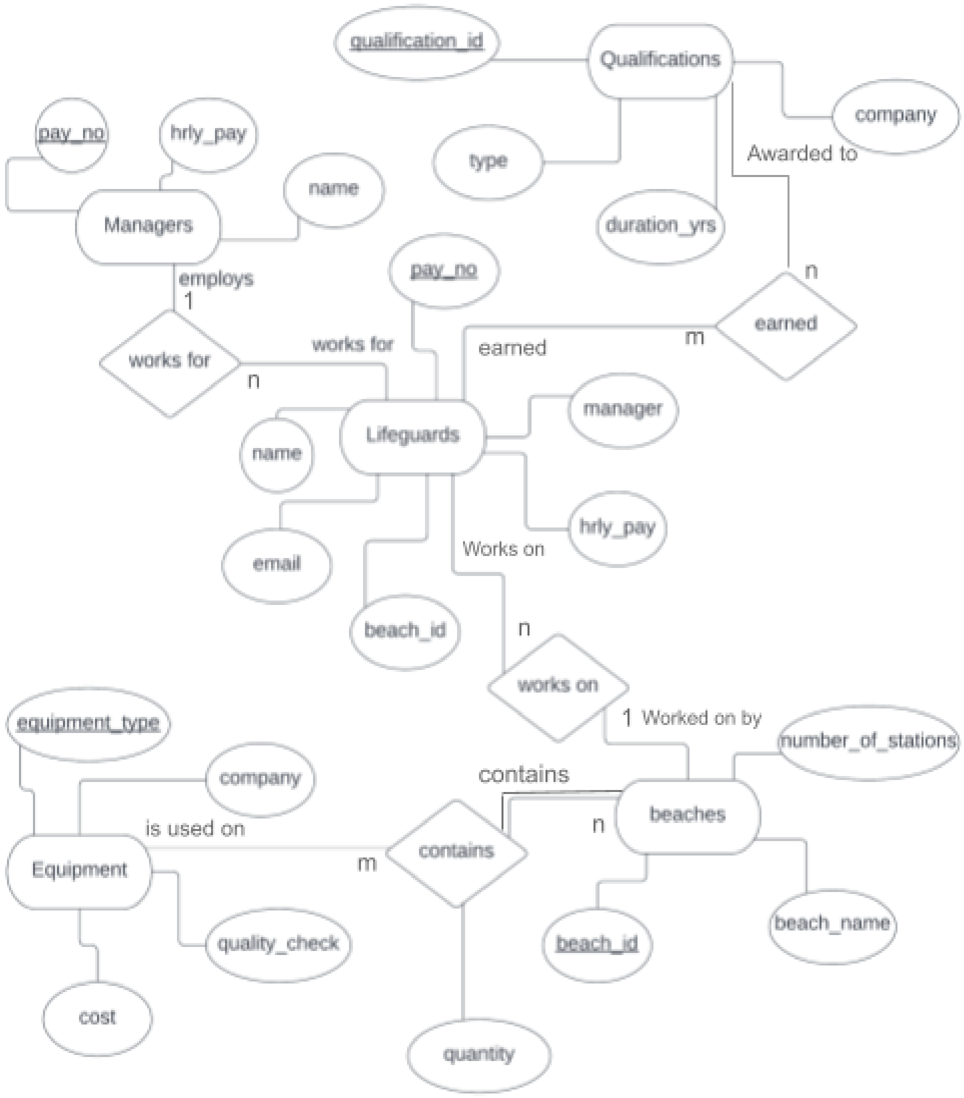
The equipment table has all the equipment that might be needed on a beach. It has an equipment type, the name of the company where the piece of equipment was bought, the cost of the equipment and the last time there was a quality check on each of the types.

Each lifeguard has a pay number, a manager, the beach they work on, their name, an email account, and the amount they are paid per hour.

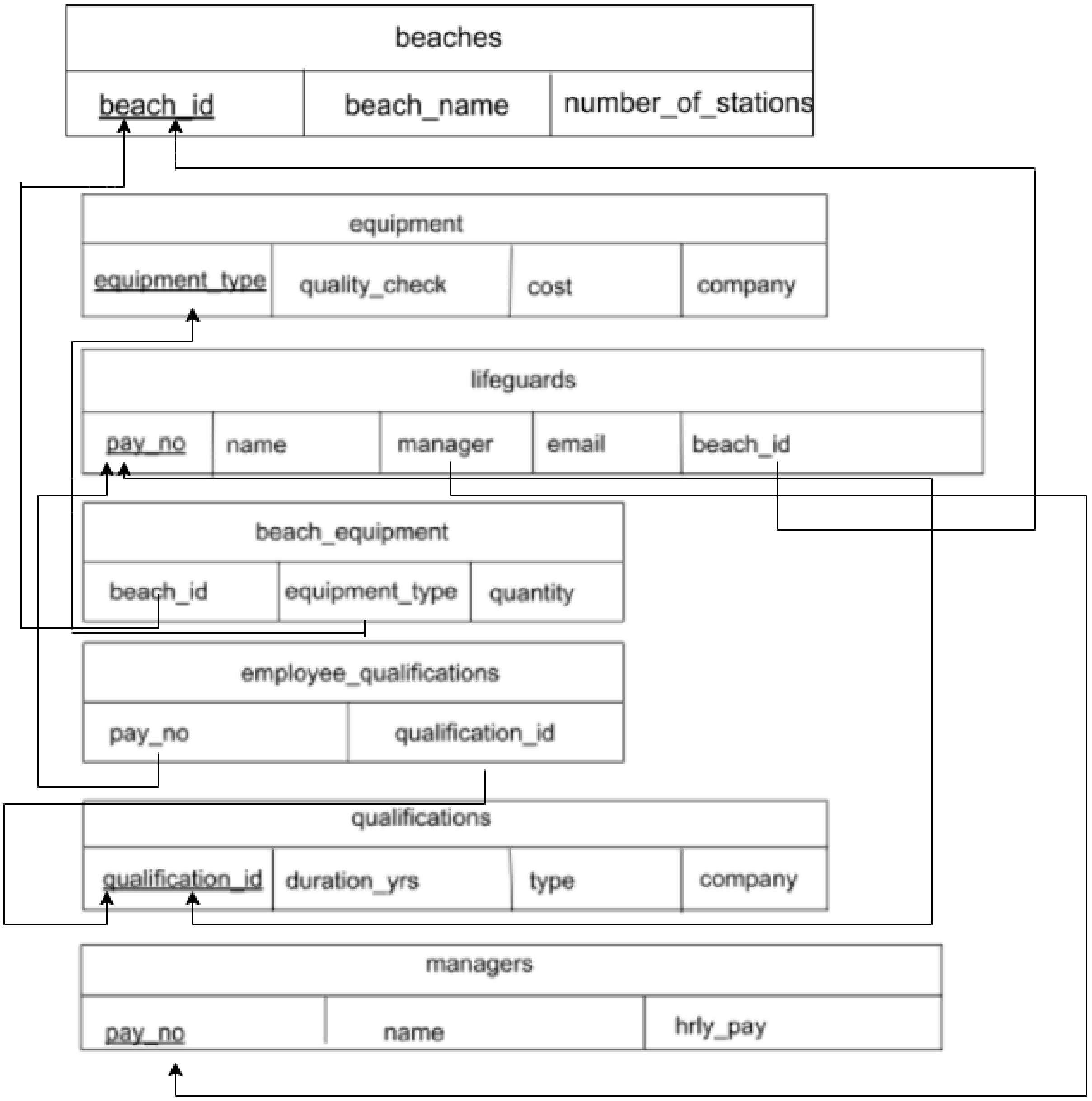
The employee qualifications table contains the qualifications each specific lifeguard has with a pay number and a qualification id.

There is also a table containing the equipment each beach has using the beach id and the equipment type. This also tells us the quantity of that type of equipment stored on the beach.

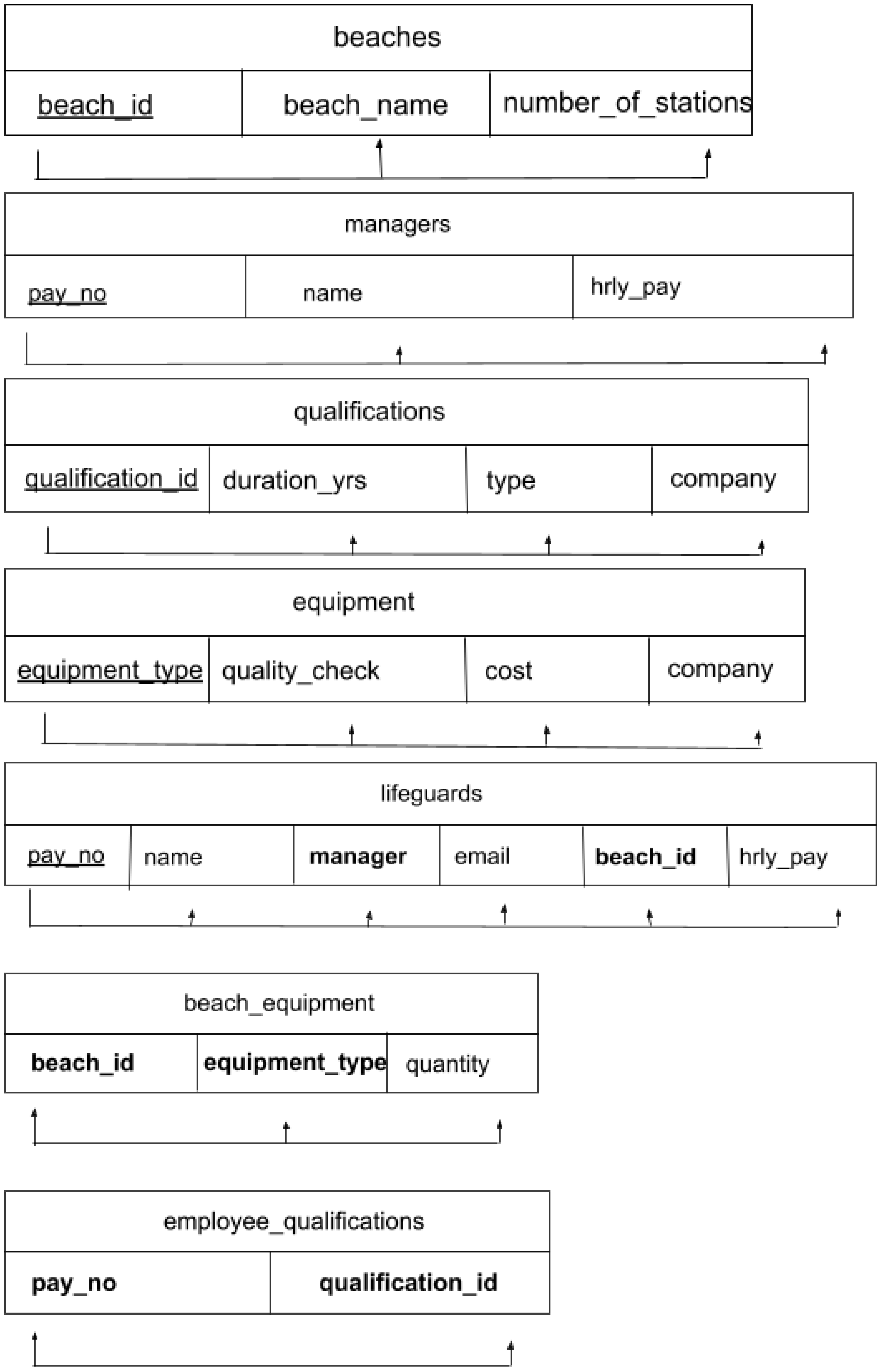
Entity Relationship Diagram



Relationship Schema



Functional Dependency Diagrams



In this diagram each primary key is underlined and foreign keys are in bold.

Section B

Creating a Database Table

```
CREATE TABLE lifeguards (
    pay_no INT PRIMARY KEY,
    name varchar(255) NOT NULL,
    beach_id INT,
    hrly_pay DECIMAL (7,2) NOT NULL,
    manager INT,
    email varchar(255) NOT NULL,
    FOREIGN KEY (beach_id) REFERENCES beaches (beach_id),
    FOREIGN KEY (manager) REFERENCES managers (pay_no)
);
```

This piece of code shows the creation of the lifeguards table. The pay number is an integer which is the primary key so it cannot be null and it must be unique. The name of the lifeguard is up to 255 characters and it cannot be null as every lifeguard must have a name. The beach_id for each lifeguard is an integer and this is a foreign key that references the primary key in the beaches table. This can be null as some lifeguards may not be assigned to a beach. Each lifeguard has an hourly pay which is a decimal (for accuracy) which cannot be null. The manager is an integer which is a foreign key which references the pay number in the managers table. This can be null as not every lifeguard must have a manager. Finally, the lifeguard has an email address which cannot be null as the lifeguard must be able to be contacted.

Altering Tables

```
-- this adds a column date_started to the lifeguards table --
ALTER TABLE lifeguards
ADD date_started DATE
;

-- this adds a column phone_no to the managers table and ensures it is 10 digits --
ALTER TABLE managers
ADD phone_no INT CHECK(length(phone_no) = 10)
;
```

These two alter table commands add columns to tables lifeguards and managers. The date_started column is added to the lifeguards table and it is of type date. The phone_number column is added to the managers table and it is an integer that must be ten digits.

Trigger Operations

```
DELIMITER $$
CREATE TRIGGER update_pay AFTER INSERT ON employee_qualifications
        FOR EACH ROW BEGIN
        UPDATE lifeguards SET lifeguards.hrly_pay = lifeguards.hrly_pay + 0.20 WHERE
pay_no = new.pay_no;
```

```
   UPDATE managers SET managers.hrly_pay = managers.hrly_pay + 0.20 WHERE
pay_no = new.pay_no;
         END$$


DELIMITER $$
CREATE TRIGGER update_equipment BEFORE DELETE ON equipment
         FOR EACH ROW BEGIN
         DELETE FROM beach_equipment WHERE equipment_type =
old.equipment_type;
   END$$
```

The first trigger operation I have, “update_pay”, updates the pay of a lifeguard whenever they earn a new qualification. When the trigger is fired the pay of the lifeguard that earned the qualification is increased by 20 cents. The pay is also updated in the managers table if the lifeguard is also a manager.

The second trigger operation, “update_equipment”, ensures that when a piece of equipment is removed from the “equipment” table, it is also removed from the beach_equipment table. This makes sure that the equipment for each beach is up to date with the equipment that the council has.

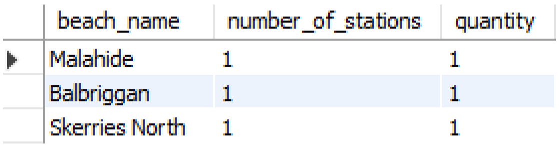
Creating of Views

```
DROP VIEW IF EXISTS boat_beaches;

CREATE VIEW boat_beaches AS
SELECT beach_name, number_of_stations, quantity
FROM beach_equipment, beaches
WHERE beach_equipment.equipment_type = 'Power Boat' AND
beach_equipment.beach_id = beaches.beach_id
;
```

This view selects the beach name the number of stations and the quantity of power boats it has. It takes this information from the beach equipment and beaches tables. It only takes the information where the equipment type is power boat and the beach id is the same in both tables to ensure that it gives the correct information.

This view yields the following table:



beach_name	number_of_stations	quantity
Malahide	1	1
Balbriggan	1	1
Skerries North	1	1

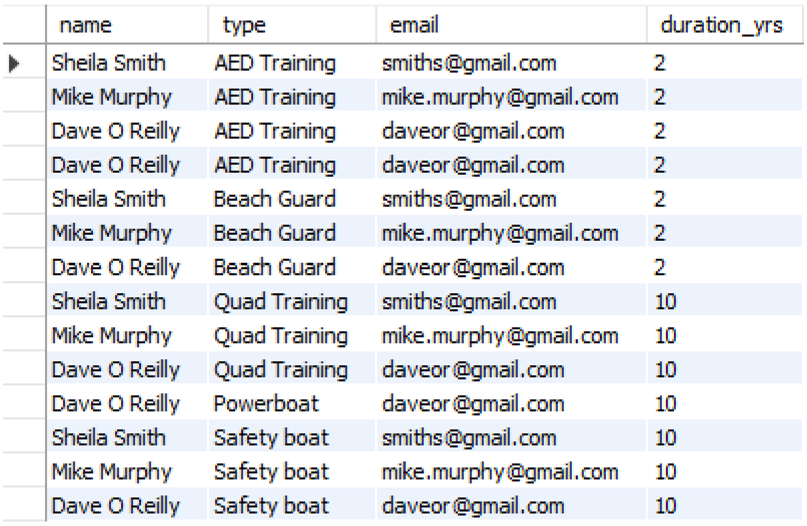
```
DROP VIEW IF EXISTS manager_qualifications;

CREATE VIEW manager_qualifications AS
```

```
SELECT lifeguards.name, qualifications.type, email, duration_yrs
FROM managers, qualifications, lifeguards, employee_qualifications
WHERE managers.pay_no = lifeguards.pay_no AND lifeguards.pay_no =
employee_qualifications.pay_no AND employee_qualifications.qualification_id =
qualifications.qualification_id
;
```

This view returns the qualifications that each manager that is also a lifeguard has. It selects from four tables. Where the lifeguard is also a manager returns the name type of qualification, email and the duration of the qualification.

This view yields the following table:



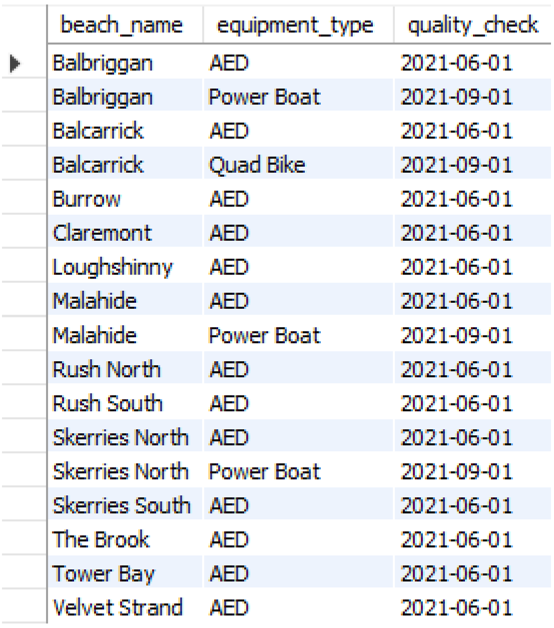
name	type	email	duration_yrs
Sheila Smith	AED Training	smiths@gmail.com	2
Mike Murphy	AED Training	mike.murphy@gmail.com	2
Dave O Reilly	AED Training	daveor@gmail.com	2
Dave O Reilly	AED Training	daveor@gmail.com	2
Sheila Smith	Beach Guard	smiths@gmail.com	2
Mike Murphy	Beach Guard	mike.murphy@gmail.com	2
Dave O Reilly	Beach Guard	daveor@gmail.com	2
Sheila Smith	Quad Training	smiths@gmail.com	10
Mike Murphy	Quad Training	mike.murphy@gmail.com	10
Dave O Reilly	Quad Training	daveor@gmail.com	10
Dave O Reilly	Powerboat	daveor@gmail.com	10
Sheila Smith	Safety boat	smiths@gmail.com	10
Mike Murphy	Safety boat	mike.murphy@gmail.com	10
Dave O Reilly	Safety boat	daveor@gmail.com	10

```
DROP VIEW IF EXISTS out_of_date_checks;

CREATE VIEW out_of_date_checks AS
SELECT beach_name, equipment.equipment_type, equipment.quality_check
FROM beaches, beach_equipment, equipment
WHERE beaches.beach_id = beach_equipment.beach_id AND
equipment.equipment_type = beach_equipment.equipment_type AND
equipment.quality_check < '2022-01-01'
ORDER BY beach_name
;
```

This view returns the beaches and the equipment they have that requires a quality check. It will return any equipment that has been checked before 2022. It also orders by name.

This view yields the following result:



beach_name	equipment_type	quality_check
Balbriggan	AED	2021-06-01
Balbriggan	Power Boat	2021-09-01
Balcarrick	AED	2021-06-01
Balcarrick	Quad Bike	2021-09-01
Burrow	AED	2021-06-01
Claremont	AED	2021-06-01
Loughshinny	AED	2021-06-01
Malahide	AED	2021-06-01
Malahide	Power Boat	2021-09-01
Rush North	AED	2021-06-01
Rush South	AED	2021-06-01
Skerries North	AED	2021-06-01
Skerries North	Power Boat	2021-09-01
Skerries South	AED	2021-06-01
The Brook	AED	2021-06-01
Tower Bay	AED	2021-06-01
Velvet Strand	AED	2021-06-01

Populating Tables

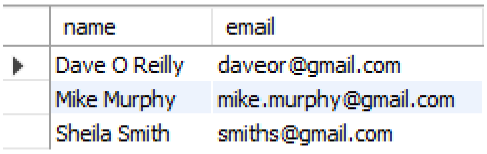
```
INSERT INTO equipment (equipment_type, company, quality_check, cost)
VALUES
    ('First Aid Kit', 'sports.ie', '2022-01-01', 37.95),
    ('AED', 'Philips', '2021-06-01', 1500.00),
    ('Rescue Board', 'Ishka Sports', '2022-01-01', 1199.00),
    ('Quad Bike', 'The Visor Shop', '2021-09-01', 3962.78),
    ('Power Boat', 'Boats.ie', '2021-09-01', 7500.50)
;
```

This is the populating of the “equipment” table. There is an insert into the “equipment” table and each of the columns is inserted into. A list of values containing each of the types of equipment is added: 'First Aid Kit', 'AED’, ‘Rescue Board’, ‘Quad Bike’, ‘Power Boat’. In the second column, the company that provided each piece of equipment is inserted. There is also the date of the last quality check of each type of equipment and finally, the cost of the equipment is inserted.

Retrieving Information from the Database

```
-- gets the email of each lifeguard that is also a manager as well as their name --
SELECT lifeguards.name, lifeguards.email
FROM managers
INNER JOIN lifeguards ON managers.pay_no = lifeguards.pay_no
;
```

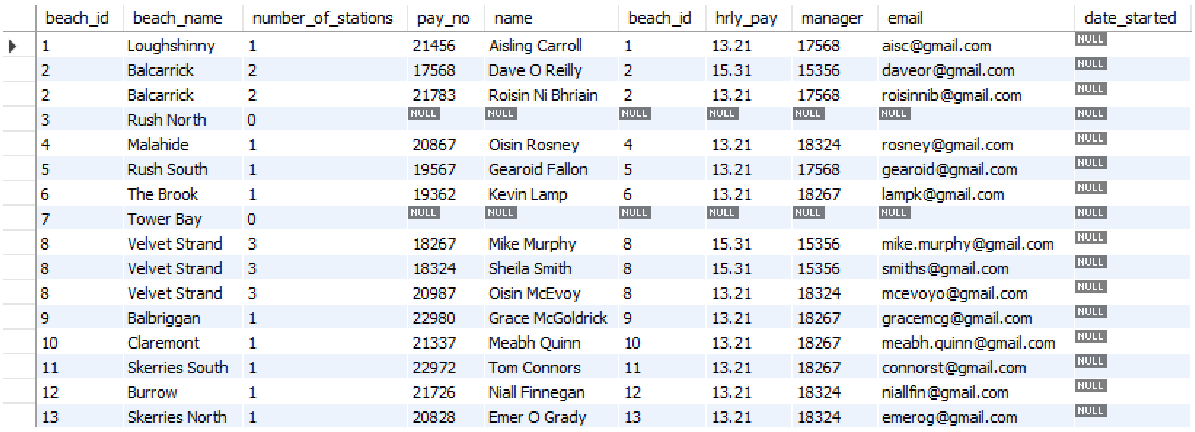
This will retrieve the emails of all lifeguards that are also a manager as well as their name using an inner join.



name	email
Dave O Reilly	daveor@gmail.com
Mike Murphy	mike.murphy@gmail.com
Sheila Smith	smiths@gmail.com

```
-- this shows all the information on the beaches and the lifeguards who work on them --
SELECT *
FROM beaches
LEFT JOIN lifeguards ON lifeguards.beach_id = beaches.beach_id
;
```

This piece of code will do a left join and return the information on all the lifeguards and the beaches they work on.



beach_id	beach_name	number_of_stations	pay_no	name	beach_id	hrly_pay	manager	email	date_started
1	Loughshinny	1	21456	Aisling Carroll	1	13.21	17568	aisc@gmail.com	NULL
2	Balcarrick	2	17568	Dave O Reilly	2	15.31	15356	daveor@gmail.com	NULL
2	Balcarrick	2	21783	Roisin Ni Bhriain	2	13.21	17568	roisinnib@gmail.com	NULL
3	Rush North	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL
4	Malahide	1	20867	Oisin Rosney	4	13.21	18324	rosney@gmail.com	NULL
5	Rush South	1	19567	Gearoid Fallon	5	13.21	17568	gearoid@gmail.com	NULL
6	The Brook	1	19362	Kevin Lamp	6	13.21	18267	lampk@gmail.com	NULL
7	Tower Bay	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL
8	Velvet Strand	3	18267	Mike Murphy	8	15.31	15356	mike.murphy@gmail.com	NULL
8	Velvet Strand	3	18324	Sheila Smith	8	15.31	15356	smiths@gmail.com	NULL
8	Velvet Strand	3	20987	Oisin McEvoy	8	13.21	18324	mcevoyo@gmail.com	NULL
9	Balbriggan	1	22980	Grace McGoldrick	9	13.21	18267	gracemcg@gmail.com	NULL
10	Claremont	1	21337	Meabh Quinn	10	13.21	18267	meabh.quinn@gmail.com	NULL
11	Skerries South	1	22972	Tom Connors	11	13.21	18267	connorst@gmail.com	NULL
12	Burrow	1	21726	Niall Finnegan	12	13.21	18324	niallfin@gmail.com	NULL
13	Skerries North	1	20828	Emer O Grady	13	13.21	18324	emerog@gmail.com	NULL

Security Commands

```
DROP ROLE IF EXISTS 'manager';
CREATE ROLE 'manager';

GRANT ALL PRIVILEGES
ON council
TO 'manager'
;

DROP ROLE IF EXISTS 'lifeguard';
CREATE ROLE 'lifeguard';

GRANT SELECT
ON council.beaches
TO 'lifeguard'
;

GRANT SELECT
ON council.equipment
TO 'lifeguard'
;

GRANT SELECT
```

```
ON council.qualifications
TO 'lifeguard'
;
```

This section of code creates two roles: manager and lifeguard. Manager is given all privileges on the database as they should be allowed to insert, delete and view things on all tables. Lifeguard is only allowed to view the beaches, equipment, and qualifications tables as there is sensitive information such as the hourly pay in the lifeguard tables and some of the other tables as well.