SQL Sandbox SOLUTIONS

Task 1:

Using the Train_journeys table output all departure times of trains going to London St Pancras using a SELECT statement.

```
SELECT departure from train_journeys
WHERE destination = 'London St Pancras';
```

Task 2:

If you couldn't already tell, the Train_journeys table is a bit messy and has many unnecessary columns and rows which we need to get rid of.

a) First we should remove the unnecessary columns using the ALTER statement. Unnecessary columns include - columns that repeat data from another column and columns that are not relevant to the table.

b) Next we need to remove unnecessary rows of data using a DELETE statement. Unnecessary rows include - rows that repeat data from another row and rows that have too much missing data.

a)
```
ALTER TABLE train_journeys
DROP COLUMN final_stop;

ALTER TABLE train_journeys
DROP COLUMN leaving_time;

ALTER TABLE train_journeys
DROP COLUMN train_colour;
```

b)
```
DELETE FROM train_journeys
WHERE journey_id = 2;

DELETE FROM train_journeys
WHERE departure is NULL;
```

Task 3:

To further clean the database we need to make sure the data is consistent across the table. Train_journeys has a mix of data that is both capitalised and uncapitalised which needs to be fixed using the UPDATE statement.

```
UPDATE train_journeys
SET destination = 'Manchester picadilly'
WHERE destination = 'manchester picadilly';

UPDATE train_journeys
SET destination = 'Liverpool'
WHERE destination = 'liverpool';
```

Task 4:

Now Train_journeys has been cleaned to an acceptable degree we are going to add a passenger table using a CREATE statement.

This table must include:

- Columns: email, first_name, phone_no and journey_id
- A foreign key that connects the journey_id in this table to the one in Train_journeys

```
CREATE TABLE passenger(
  email varchar(30),
  first_name varchar(20),
  phone_no varchar(11),
  journey_id int
  FOREIGN KEY (journey_id) REFERENCES train_journeys(journey_id)
 );
```

Task 5:

Now our table is created and adds at least one row of appropriate values using an INSERT INTO statement.

You can put any values you want for this task but this is the general format:

```
INSERT INTO passenger
VALUES
('t.carbonell@email.com','Tomas','07888888888',1);
```

Task 6:

Finally with both tables connected we can retrieve all data at once using JOIN statements. Your task is to output a single passenger's email, first name, phone number, destination and departure using a JOIN statement.

There are multiple ways you can join the two tables together but this is the solution I came up with first:

```
SELECT email,first_name,phone_no,departure,destination
FROM passenger
LEFT JOIN train_journeys
ON passenger.journey_id = train_journeys.journey_id;
```