|  |  |
| --- | --- |
| **MODULE 2 -- PRACTICAL EXERCISES** | |
| **SLIDE 7**  Using the table ‘parts’, return all unique part names. What happens if we want to return all unique parts and their id number? Why? | USE PARTS;  SELECT DISTINCT P.PNAME  FROM PART P;  and  SELECT DISTINCT P.P\_ID, P.PNAME  FROM PART P; |
| **SLIDE 7**  Refer to the table ‘projects’ and return all projects that are run in London | USE PARTS;  SELECT DISTINCT  J.J\_ID, J.JNAME, J.CITY  FROM  PROJECT J  WHERE  J.CITY = 'London'; |
| **SLIDE 16**  Design and create a relational normalised database called **customers**.  Set reasonable **primary keys** to the tables.  Set NOT NULL **constraints** on the columns that you think must have values. | CREATE DATABASE customers;  USE customers;  CREATE TABLE customer (  customer\_id INTEGER NOT NULL,  first\_name VARCHAR(55) NOT NULL,  last\_name VARCHAR(55) NULL,  CONSTRAINT PK\_customer PRIMARY KEY (customer\_id)  );  CREATE TABLE address (  address\_id INTEGER NOT NULL,  building\_number VARCHAR(55) NOT NULL,  street VARCHAR(55) NOT NULL,  city VARCHAR(55),  post\_code VARCHAR(55) NOT NULL,  country VARCHAR(55),  CONSTRAINT PK\_Address PRIMARY KEY (address\_id)  );  CREATE TABLE email\_address (  email\_address\_id INTEGER NOT NULL,  email\_address\_customer\_id INTEGER,  email\_address VARCHAR(55) NOT NULL,  CONSTRAINT PK\_email\_address PRIMARY KEY (email\_address\_id)  );  CREATE TABLE phone\_number (  phone\_number\_id INTEGER NOT NULL,  phone\_number\_customer\_id INTEGER NOT NULL,  phone\_number VARCHAR(55) NOT NULL,  CONSTRAINT PK\_phone\_number PRIMARY KEY (phone\_number\_id)  );  CREATE TABLE orders (  order\_id INTEGER NOT NULL,  orders\_customer\_id INTEGER NOT NULL,  order\_date DATE NOT NULL,  CONSTRAINT PK\_order\_id PRIMARY KEY (order\_id)  ); |
| **SLIDE 27**  Add some data to the tables in the **customers database** | INSERT INTO address  (address\_id, building\_number, street, city, post\_code, country)  VALUES  (1, '20', 'Birch Alley', 'London', 'SE24 0AB', 'UK'),  (2, '17', 'Oak Street', 'London', 'SE25 0XY', NULL);  INSERT INTO customer  (customer\_id, first\_name, last\_name)  VALUES  (1, 'Jon', 'Flanders'),  (2, 'Sam', 'Smith');  INSERT INTO email\_address  (email\_address\_id, email\_address\_customer\_id, email\_address)  VALUES  (1, 2, 'ssmith@mail.com'),  (2, 1, 'jon@mail.com');      INSERT INTO phone\_number  (phone\_number\_id, phone\_number\_customer\_id, phone\_number)  VALUES  (1, 1, '555-1212'),  (2, 2, '555-3344');      INSERT INTO orders  (order\_id, orders\_customer\_id, order\_date)  VALUES  (1, 1, '2019-08-20'),  (2, 2, '2019-03-15'); |

|  |  |
| --- | --- |
| **SLIDE 27**  Alter tables **email\_address** and **phone\_number** in the **customers database** by adding **Foreign keys** that reference **Primary keys** from relevant tables.  *NB: once you have added FKs to the table use workbench tools to inspect the table in the object browser to demonstrate that all constraints are in place* | ALTER TABLE email\_address  ADD CONSTRAINT  FK\_email\_address\_customer  FOREIGN KEY(email\_address\_customer\_id)  REFERENCES customer (customer\_id);  ALTER TABLE phone\_number  ADD CONSTRAINT FK\_phone\_number\_customer  FOREIGN KEY(phone\_number\_customer\_id)  REFERENCES customer (customer\_id); |