| **Database Development: SQL used and explained** | |
| --- | --- |
| **List of Tables in our Database** | |
| **Book**  **Customer**  **Customer\_Management**  **Customer\_Phone\_No**  **Order** | **Order\_Book**  **Review**  **Stock**  **Supplier** |
| **Customer table - Conor Clyde** | |
|  | |
| **Data Definition Language used to create table structure** | |
| DROP TABLE IF EXISTS `customer`;  CREATE TABLE IF NOT EXISTS `customer`  (  `Customer\_ID` int(11) NOT NULL AUTO\_INCREMENT,  `Forename` varchar(40) DEFAULT NULL,  `Surname` varchar(40) DEFAULT NULL,  `Address` varchar(80) DEFAULT NULL,  `Postcode` varchar(8) DEFAULT NULL,  PRIMARY KEY (`Customer\_ID`)  ) ENGINE=InnoDB AUTO\_INCREMENT=29 DEFAULT CHARSET=latin1; | |
| **SQL Explanation:**   * Checks if a table called customer exists * If it exists, delete it. Then create a table called customer * If it does not exist, there is no table to delete so only create a table called customer * Within the brackets, the columns to be created within the customer table are specified * Columns to be created are Customer\_ID, Forename, Surname, Address, Postcode * Following the column names, the type of data that is to be held in each column is defined * Int refers to an integer.Varchar(40) refers to text of length 40 characters. * NOT NULL specifies that the column must have a value for every entry in the table. * Auto\_INCREMENT tells the column to automatically create a value incremented from the previous entry in the table * DEFAULT specifies the value that is set automatically if no value is set. In this case, it is set to NULL. * Customer ID is set as the primary key of the table - It uniquely identifies each entry in the table * The engine is specified to be InnoDB | |
| **Data Manipulation Language used to populate table** | |
| INSERT INTO `customer` (`Customer\_ID`, `Forename`, `Surname`, `Address`, `Postcode`)  VALUES  (1, 'Mary', 'Stewart', '122 Train St', 'PA1G 1YQ'),  (2, 'David', 'Brown', '117 Blue Street', 'BT12 3RT'),  (3, 'John', 'White', '112 Green Avenue', 'BT11 7RT'),  (4, 'Sarah', 'Lee', '150 King\'s Street', 'BT48 8DX'),  (5, 'Jack', 'Brand', '180 Main Street', 'GF5X 2ES'),  (6, 'Matthew', 'Williams', '70 Rosewood Avenue', 'BT47 3JF'),  (7, 'Julie', '1912 Kimberly Way', '18 Tain St, Gourock', 'BT48 3JE'),  (8, 'Mike', 'Green', '10 Main Road', 'BT47 3MF'),  (9, 'Owen', 'Smith', '43 Hollyfoot Hill', 'BT47 4FB'),  (10, 'Will', 'Jones', '20 Jubilee Terrace, Gourock', 'RH74 1LE'),  (11, 'Sally', 'Mullan', '6 Scotney Close', 'MD31 9JB'),  (12, 'Nicole', 'Taylor', '4107 Robinson Lane', 'BT48 7IY'),  (13, 'Steve', 'Johnson', '145 Main Street', 'DB1B 3FF'),  (14, 'Steven', 'Evans', '9 Second Avenue', 'DB4F 3QG'),  (15, 'David', 'Wilson', '170 Oak Road', 'DB3B 3TG'),  (16, 'Shea', 'Conway', '12 Main St', 'BT723TR'),  (17, 'Bob', 'Adams', '23 Low Avenue', 'BT678RY'),  (18, 'Greg', 'Brown', '31 Leaf Avenue', 'BT546TR'),  (19, 'Katy', 'Conor', '72 Cam Road', 'BT723PR'),  (20, 'Derek', 'Conway', '1 Great Lane', 'GIR 0AA'),  (21, 'Derek', 'Conway', '1 Great Lane', 'GIR 0AA'),  (22, 'John', 'Philips', '4 Long Lane', 'BT76 9PR'),  (23, 'Daren', 'McNally', '16 Near Road', 'BT79 3PT'),  (24, 'Andy', 'Mul', '2 Cree Lane', 'BT78 9PR'),  (25, 'Greg', 'Marley', '64 Zoo Lane', 'BT79 3PT'),  (26, 'Trevor', 'Smith', '88 West Street', 'BT5 4TR'),  (27, 'Shiela', 'Thomson', '6 Highfield Road', 'BT1 5PO'),  (28, 'Clare', 'Ward', '164 Windsor Road', 'BT79 9PR'); | |
| **SQL Explanation:**   * The SQL shown populates the Customer table with data * INSERT INTO customer specifies the data to be inserted into the customer table * The columns to insert data into are specified * The data to be inserted into the specified columns are stated after VALUE * There are many rows because many rows of data are being added to the table | |
| **Customer\_Management Table - Conor Clyde** | |
|  | |
| **Data Definition Language used to create table structure** | |
| DROP TABLE IF EXISTS `customer\_management`;  CREATE TABLE IF NOT EXISTS `customer\_management`  (  `Customer\_ID` int(11) NOT NULL AUTO\_INCREMENT,  `Email` varchar(200) NOT NULL,  `Password` varchar(80) NOT NULL,  PRIMARY KEY (`Customer\_ID`),  UNIQUE KEY `Email` (`Email`)  ) ENGINE=InnoDB AUTO\_INCREMENT=29 DEFAULT CHARSET=latin1; | |
| **SQL Explanation:**   * Checks if customer\_management table exists and deletes it if it does * Creates table called customer\_management * Specifies the columns to be created within the table (customer\_id, email, password) * Specifies the types of data to be entered into these columns - int for integer, and varchar for text * Set customer\_id as the primary key * Sets email as unique key - Must be unique to customer * Sets Engine to InnoDB | |
| **Data Definition Language used to link table with other tables in database** | |
| ALTER TABLE `customer\_management`  ADD CONSTRAINT ```fk\_Customer\_ID` FOREIGN KEY (`Customer\_ID`) REFERENCES `customer` (`Customer\_ID`); | |
| **SQL Explanation:**   * Foreign key constraint is added * This constraint establishes a relationship between the customer\_management table and the customer table by including a column in the customer\_management table which links a column in the customer table * The column linking the two tables is customer\_id, which is the primary key of the customer table | |
| **Data Manipulation Language used to populate table** | |
| INSERT INTO `customer\_management` (`Customer\_ID`, `Email`, `Password`)  VALUES  (1, 'Mary@email.com', 'TestPassword'),  (2, 'David@email.com', 'TestPassword'),  (3, 'John@email.com', 'TestPassword'),  (4, 'Sarah@email.com', 'TestPassword'),  (5, 'Jack@email.com', 'TestPassword'),  (6, 'Matthew@email.com', 'TestPassword'),  (7, 'Julie@email.com', 'TestPassword'),  (8, 'Mike@email.com', '1234'),  (9, 'Owen@email.com', 'TestPassword'),  (10, 'Will@email.com', 'TestPassword'),  (11, 'Sally@email.com', 'TestPassword'),  (12, 'Nicole@email.com', 'TestPassword'),  (13, 'Steve@email.com', 'TestPassword'),  (14, 'Steven@email.com', 'TestPassword'),  (15, 'GregD@email.com', 'TestPassword123'),  (16, 'sheacon@hotmail.com', 'Abc123'),  (17, 'bob@outlook.com', 'Qwerty6'),  (18, 'grew@outllook.com', 'Carrot11'),  (19, 'katy23@outllook.co.uk', 'keg432'),  (20, 'derek@hotmail.co.uk', 'Qwe123'),  (22, 'John23@hotmail.co.uk', 'Bob1234'),  (23, 'daren@outlook.cod', 'Wasd12'),  (24, 'andy43@outlook.com', 'Werer3434'),  (25, 'greg3222@gmail.com', 'Qsd1212'),  (26, 'Trev23@gmail.co.uk', 'Password123'),  (27, 'Shiela5@gmail.com', 'Tank935'),  (28, 'clare9@gmail.com', 'Pass1000'); | |
| **SQL Explanation:**   * Populates the customer\_management table with data * Specifies that data is to be inserted into the customer\_management table and which columns to add the data to * The data to be added is stated after VALUEs * There are many rows of data being added to the table | |
| **Customer\_Phone\_No table - Conor Clyde** | |
|  | |
| **Data Definition Language used to create table structure** | |
| DROP TABLE IF EXISTS `customer\_phone\_no`;  CREATE TABLE IF NOT EXISTS `customer\_phone\_no`  (  `Customer\_ID` int(11) NOT NULL AUTO\_INCREMENT,  `Country\_Code` varchar(10) NOT NULL,  `Phone\_No` varchar(24) NOT NULL,  `Type` varchar(40) DEFAULT NULL,  `Comment` varchar(80) DEFAULT NULL,  PRIMARY KEY (`Customer\_ID`,`Country\_Code`,`Phone\_No`)  ) ENGINE=InnoDB AUTO\_INCREMENT=29 DEFAULT CHARSET=latin1; | |
| **SQL Explanation:**   * Checks if a table called customer\_phone\_no exists. If it does, delete it. * Creates a new table called customer\_phone\_no * Specifies the names of the columns to be created within the table and the types of data that they can hold - int for integer, varchar for text * Specifies a composite primary key (primary key consisting of multiple columns) to uniquely identify each row within the table * Specifies the engine to be InnoDB | |
| **Data Definition Language used to link table with other tables in database** | |
| ALTER TABLE `customer\_phone\_no`  ADD CONSTRAINT `fk\_Customer\_To\_Customer\_Phone\_No` FOREIGN KEY (`Customer\_ID`) REFERENCES `customer` (`Customer\_ID`) ON DELETE CASCADE ON UPDATE CASCADE; | |
| **SQL Explanation:**   * Establishes relationship between customer\_phone\_no table and customer table by placing a foreign key restraint on the customer\_id column in the customer\_phone\_no table which allows it to reference the customer\_id column in the customer table | |
| **Data Manipulation Language used to populate table** | |
| INSERT INTO `customer\_phone\_no` (`Customer\_ID`, `Country\_Code`, `Phone\_No`, `Type`, `Comment`)  VALUES  (1, '+44', '7518445353', 'Mobile', ''),  (2, '+44', '7518635338', 'Mobile', ''),  (3, '+44', '7567845533', 'Mobile', ''),  (4, '+44', '7564545464', 'Mobile', ''),  (5, '+44', '7567328654', 'Mobile', ''),  (6, '+44', '7512345533', 'Mobile', ''),  (7, '+44', '7557532334', 'Mobile', ''),  (8, '+49', '7543543222', 'Home', ''),  (9, '+44', '7545663232', 'Mobile', ''),  (10, '+1', '7543424242', 'Mobile', ''),  (11, '+44', '7544535383', 'Home', ''),  (12, '+44', '7743533332', 'Mobile', ''),  (13, '+49', '7575434332', 'Mobile', ''),  (14, '+44', '7375454442', 'Mobile', ''),  (15, '+33', '7542312343', 'Mobile', ''),  (16, '+44', '2856745632', 'Home', ''),  (17, '+39', '2875645823', 'Mobile', ''),  (18, '+44', '2984532165', 'Mobile', ''),  (19, '+44', '9345673421', 'Mobile', ''),  (20, '+44', '2354677134', 'Mobile', ''),  (21, '+33', '2354677134', 'Mobile', ''),  (22, '+44', '3265476533', 'Mobile', ''),  (23, '+33', '2765444623', 'Mobile', ''),  (24, '+44', '3246588730', 'Mobile', ''),  (25, '+44', '3243423339', 'Mobile', ''), | |
| **SQL Explanation:**   * Specifies data is to be inserted into customer\_phone\_no table * Specifies columns to insert data into * Specifies data to be inserted into the columns after VALUES | |
| **Order-Book table - Conor Clyde** | |
|  | |
| **Data Definition Language used to create table structure** | |
| DROP TABLE IF EXISTS `order\_book`;  CREATE TABLE IF NOT EXISTS `order\_book`  (  `Order\_ID` int(11) NOT NULL,  `Book\_ID` int(11) NOT NULL,  `Quantity` int(11) NOT NULL,  PRIMARY KEY (`Order\_ID`,`Book\_ID`),  KEY `Book\_ID` (`Book\_ID`)  ) ENGINE=InnoDB DEFAULT CHARSET=latin1; | |
| **SQL Explanation:**   * Checks if a table called order\_book exists. * If it exists, delete it. * Creates a new table called order\_book * Specifies the names of the columns to be created within order\_book * Specifies the data type allowed for the value of each column - int for integer * Creates a composite primary key of order\_id and book\_id - Used to uniquely identify each row in the table * Specifies the engine as InnoDB | |
| **Data Definition Language used to link table with other tables in database** | |
| ALTER TABLE `order\_book`  ADD CONSTRAINT `fk\_Book\_To\_Order\_Book` FOREIGN KEY (`Book\_ID`) REFERENCES `book` (`Book\_ID`) ON DELETE CASCADE ON UPDATE CASCADE,  ADD CONSTRAINT `fk\_Order\_To\_Order\_Book` FOREIGN KEY (`Order\_ID`) REFERENCES `order` (`Order\_ID`) ON DELETE CASCADE ON UPDATE CASCADE; | |
| **SQL Explanation:**   * Establishes relationship between order\_book and book table by placing a foreign key restraint on the book\_id column in the order\_book table which allows it to reference the book\_id column in the book table * Establishes relationship between order\_book and order table by placing a foreign key restraint on the order\_id column in the order\_book table which allows it to reference the order\_id column in the order table | |
| D**ata Manipulation Language used to populate table** | |
| INSERT INTO `order\_book` (`Order\_ID`, `Book\_ID`, `Quantity`)  VALUES  (101, 2, 1),  (101, 3, 1),  (102, 1, 1),  (103, 1, 1),  (103, 2, 1),  (103, 10, 4),  (104, 2, 8),  (105, 3, 1),  (106, 3, 2); | |
| **SQL Explanation:**   * Specifies data is to be inserted into order\_book table * Specifies columns to insert data into * Specifies data to be inserted into the columns after VALUES | |
| **Book table-Stephen McNally** | |
| **Data Definition Language used to create table structure** | |
| DROP TABLE IF EXISTS `book`;  CREATE TABLE IF NOT EXISTS `book` (  `Book\_ID` int(4) NOT NULL AUTO\_INCREMENT,  `Book\_Title` varchar(100) NOT NULL,  `Book\_Desc` varchar(800) DEFAULT NULL,  `Author` varchar(50) NOT NULL,  `Price` decimal(3,2) NOT NULL,  `Genre` varchar(25) NOT NULL,  `Book\_Length` int(4) NOT NULL,  `Language` varchar(25) NOT NULL,  `ISBN-10` varchar(14) NOT NULL,  `ISBN-13` varchar(14) NOT NULL,  `Cover\_Image` varchar(80) DEFAULT NULL,  PRIMARY KEY (`Book\_ID`)  ) ENGINE=InnoDB AUTO\_INCREMENT=16 DEFAULT CHARSET=latin1; | |
| **SQL Explanation:**  The code listed above will create the table book. If this table already exists within the database it will be dropped (deleted) and replaced with this table. This table will contain a total of 11 attributes which consist of Book\_ID, Bookk\_Title, Book\_Desc, Author, Price, Genre, Book\_Length, Language, ISBN-10, ISBN-13, Cover\_Image. A primary key constraint will be added to the Book\_ID. This attribute will auto increment and will uniquely identify the records in the table. | |
| **Data Manipulation Language used to populate table** | |
| INSERT INTO `book` (`Book\_ID`, `Book\_Title`, `Book\_Desc`, `Author`, `Price`, `Genre`, `Book\_Length`, `Language`, `ISBN-10`, `ISBN-13`, `Cover\_Image`) VALUES  (1, 'Fantastic Mr. Fox', 'Three horrid farmers - Boggis, Bunce and Bean - hate cunning Mr Fox, who outwits them at every turn. But poor Mr Fox and his friends don\'t realise how determined the farmers are to get them...', 'Roald Dahl', '5.99', 'Children', 112, 'English', '0141365447', '978-0141365442', 'Resources/Images/fmf'),  (2, 'Charlie and the Chocolate Factory', 'Nobody has seen Willy Wonka - or inside his amazing chocolate factory - for years. When he announces plans to invite the winners of five Golden Tickets hidden inside the wrappers of chocolate bars to visit his factory, the whole world is after those tickets!', 'Roald Dahl', '4.99', 'Children', 208, 'English', '0141365374', '', 'Resources/Images/catcf'),  (3, 'The BFG', 'The Big Friendly Giant is unlike other giants. For a start, he doesn\'t like to eat people and it\'s not long before he becomes orphan Sophie\'s very best friend.', 'Roald Dahl', '6.00', 'Children', 224, 'English', '9780141365428', '978-0141365428', 'Resources/Images/bfg'),  (4, 'Matilda', 'Matilda Wormwood is only five years old, but she is a genius. Unfortunately her parents are too stupid to even notice. Worse, her horrible headmistress Miss Trunchbull is a bully who makes life difficult for Matilda\'s teacher, Miss Honey, and her friends. But what Miss Trunchbull doesn\'t know is that Matilda has a trick or two up her sleeve...', 'Roald Dahl', '5.99', 'Children', 256, 'English', '9780141365466', '978-0141365466', 'Resources/Images/mat'),  (5, 'The Twits', 'Mr Twit hates his wife. Mrs Twit detests her husband. They like nothing more than playing wicked tricks on one another. Sooner or later, things are going to go too far...', 'Roald Dahl', '3.99', 'Children', 112, 'English', '9780141365497', '978-0141365497', 'Resources/Images/ttw'),  (6, 'James and the Giant Peach', 'James Henry Trotter lives with his two horrid aunts, Spiker and Sponge. He hasn\'t got a single friend in the whole wide world. That is not, until he meets the Old Green Grasshopper and the rest of the insects aboard a giant, magical peach!', 'Roald Dahl', '2.99', 'Children', 176, 'English', '9780141365459', '978-0141365459', 'Resources/Images/jatgp'),  (7, 'The Enormous Crocodile', 'Crocodiles are such greedy creatures - and their favourite lunchtime snack happens to be a juicy child or two! The Enormous Crocodile isn\'t as smart as he thinks though, so he had better watch out...', 'Roald Dahl', '4.00', 'Children', 64, 'English', '014136551X', '978-0141365510', 'Resources/Images/tec'),  (8, 'Esio Trot', 'Saying things backwards can make magic happen. Just ask Mr Hoppy and Mrs Silver! Esio Trot is the story of a very shy man and a very kind woman, and a small tortoise called Alfie who brings them together.', 'Roald Dahl', '6.99', 'Children', 80, 'English', '9780141369389', '978-0141369389', 'Resources/Images/et'),  (9, 'The Magic Finger', 'When the girl in this story gets cross, strange things start happening. Above all, she can\'t bear it when people are cruel to animals. So when her neighbours the Greggs go shooting, her magic finger teaches them a lesson they\'ll never forget...', 'Roald Dahl', '3.99', 'Children', 80, 'English', '9780141365404', '978-0141365404', 'Resources/Images/tmf'),  (10, 'George\'s Marvellous Medicine', 'George\'s nasty old grandma needs teaching a lesson. George decides the best remedy for her grumpiness is a special home-made medicine. But Grandma gets more than she bargained for!', 'Roald Dahl', '4.00', 'Children', 128, 'English', '9780141365503', '978-0141365503', 'Resources/Images/gmm'),  (11, 'The Complete Adventures of Charlie and Mr Willy Wonka', 'The Complete Adventures of Charlie and Mr Willy Wonka contains two of Roald Dahl\'s most extraordinary stories. In, Charlie and the Chocolate Factory Mr Wonka opens the gates to his factory to five lucky children, and Charlie is the fifth and final Golden Ticket winner. Beyond the factory doors lies a river of chocolate, edible plants and everlasting gobstoppers. In Charlie and the Great Glass Elevator, Charlie\'s adventures with Willy Wonka continues.', 'Roald Dahl', '9.99', 'Children', 400, 'English', '0141365390', '', 'Resources/Images/tcaocamww'),  (12, 'Danny the Champion of the World', 'Danny lives with his dad in a caravan at the edge of the wood. He thinks his dad is the best father in the world. But Danny doesn\'t know everything, and even his brilliant dad has secrets...', 'Roald Dahl', '5.00', 'Children', 240, 'English', '9780141365411', '978-0141365411', 'Resources/Images/dtcotw'),  (13, 'Charlie and the Great Glass Elevator', 'Willy Wonka has asked Charlie and the rest of the Bucket family to live with him. Now, moments after Charlie and the Chocolate Factory ended, we rejoin the adventure as the Great Glass Elevator blasts into outer space...', 'Roald Dahl', '4.50', 'Children', 208, 'English', '0141365382', '978-0141365381', 'Resources/Images/catgge'),  (14, 'The Witches', 'Witches absolutely detest children. To a witch, a child smells like dogs\' droppings. And now the Grand High Witch is planning to get rid of every child in England - can anybody stop them?', 'Roald Dahl', '6.00', 'Children', 368, 'English', '0141365471', '978-0141365473', 'Resources/Images/twt'),  (15, 'Boy and Going Solo', 'Two stories in one! Boy describes an unadulterated childhood - sad and funny, macabre and delightful - that inspired Britain\'s favourite storyteller, Boy speaks of an age which vanished with the coming of the Second World War. Going Solo tells of how, when he grew up, Roald Dahl left England for Africa - and a series of daring and dangerous adventures began...\r\n\r\n', 'Roald Dahl', '8.00', 'Children', 464, 'English', '9780141365541', '978-0141365541', 'Resources/Images/bags'); | |
| **Stock table - Stephen McNally** | |
| **Data Definition Language used to create table structure** | |
| DROP TABLE IF EXISTS `stock`;  CREATE TABLE IF NOT EXISTS `stock` (  `Book\_ID` int(4) NOT NULL,  `book\_Supplier\_ID` int(11) NOT NULL,  `Stock\_Level` int(4) NOT NULL,  KEY `BookID` (`Book\_ID`),  KEY `book\_Supplier\_ID` (`book\_Supplier\_ID`)  ) ENGINE=InnoDB DEFAULT CHARSET=latin1; | |
| **SQL Explanation:**  The following SQL will drop (delete) the table called stock if it already exists within the database. After that the table stock will be created. This table will include three attributes called Book\_ID, book\_Supplier\_ID and Stock\_Level. This table uniquely identifies its records using foreign keys which it references from separate tables. This is explained in the next section. | |
| **Data Definition Language used to link table with other tables in database** | |
| ALTER TABLE `stock`  ADD CONSTRAINT `books\_ibfk\_1` FOREIGN KEY (`Book\_ID`) REFERENCES `book` (`Book\_ID`),  ADD CONSTRAINT `stock\_supplier\_ID\_fk` FOREIGN KEY (`book\_Supplier\_ID`) REFERENCES `supplier` (`supplier\_ID`); | |
| **SQL Explanation:**  The SQL code above will add foreign keys to the table stock. Unlike the primary key this was done separately to avoid errors in case the tables containing the primary keys did not exist within the database yet. The first foreign key is called books\_ibfk\_1. This will be represented by the Book\_ID attribute in the stock table which references the BookID attribute from the book table. The second foreign key is called stock\_supplier\_ID\_fk. This is represented by the book\_Supplier\_ID attribute in the stock table which references the supplier\_ID found in the supplier table. Together they form a composite key that will uniquely identify records in the stock table. | |
| **Data Manipulation Language used to populate table** | |
| INSERT INTO `stock` (`Book\_ID`, `book\_Supplier\_ID`, `Stock\_Level`) VALUES  (1, 1, 12), (2, 3, 32), (3, 4, 18),(4, 2, 27),(5, 3, 4),(6, 3, 0),(7, 4, 11),(8, 2, 43),(9, 2, 16),  (10, 1, 18),(12, 1, 33),(13, 4, 48),(14, 3, 51),(15, 1, 9),(11, 2, 10); | |
| **SQL Explanation:**  The following SQL code will insert data into the stock table. The rows where data will be entered into are as follows: Book\_ID, Book\_Supplier\_ID and Stock\_Level, in that order. A total of 15 records will be inserted into the stock table all containing different values. | |
| **Order Table** | |
|  | |
| **SQL Explanation:**  Firstly, the following sql creates the table named order and will drop any previously existing tables with the same name. The order ID field is created and set to increment automatically for each table entry. Order ids will be unique to each row and cannot be set to null. The id will also serve as the order table's primary key.  Both customer id and book id are foreign keys taken from their respective tables. These first 3 keys are all essential to the order and therefore are set to NOT-NULL. The sql then creates the following fields: Checkout date, order date, quantity, price and checkout status. The price will be influenced by the book id and the quantity as It will take the original books price from the “books table” and times it by the quantity ordered. Lasty the checkout-status will be used to determine whether an order has been paid for or is just waiting in a user's shopping car  The middle “INSERT INTO” section displays how the table is populated using sql with the exception of price which as previously mentioned is calculated using the original book price and quantity. | |
| **Data Definition Language used to create table structure** | |
| DROP TABLE IF EXISTS `order`;  CREATE TABLE IF NOT EXISTS `order` (  `OrderID` int(11) NOT NULL AUTO\_INCREMENT,  `CustomerID` int(11) NOT NULL,  `BookID` int(4) NOT NULL,  `OrderDate` date DEFAULT NULL,  `Quantity` int(11) NOT NULL,  `Price` decimal(3,2) NOT NULL,  `DeliveryDate` date DEFAULT NULL,  `Checkout-status` tinyint(1) DEFAULT NULL,  PRIMARY KEY (`OrderID`),  KEY `BookID` (`BookID`),  KEY `order\_custID\_fk2` (`CustomerID`),  KEY `order\_price\_fk` (`Price`)  ) ENGINE=InnoDB AUTO\_INCREMENT=9 DEFAULT CHARSET=latin1; | |
| **Data Definition Language used to link table with other tables in database** | |
| ALTER TABLE `order`  ADD CONSTRAINT `order\_custID\_fk2` FOREIGN KEY (`CustomerID`) REFERENCES `customer` (`Customer\_ID`),  ADD CONSTRAINT `order\_ibfk\_1` FOREIGN KEY (`BookID`) REFERENCES `books` (`BookID`) ON DELETE CASCADE ON UPDATE CASCADE,  ADD CONSTRAINT `order\_price\_fk` FOREIGN KEY (`Price`) REFERENCES `books` (`Price`);  COMMIT; | |
| **Data Manipulation Language used to populate table** | |
| INSERT INTO `order` (`OrderID`, `CustomerID`, `BookID`, `OrderDate`, `Quantity`, `Price`, `DeliveryDate`, `Checkout-status`) VALUES  (1, 7, 2, '2022-02-07', 1, '4.99', '2022-02-21', NULL),  (2, 12, 6, '2022-02-08', 3, '2.99', '2022-02-21', NULL),  (3, 4, 4, '2022-02-23', 1, '5.99', '2022-02-28', NULL),  (4, 8, 2, '2022-02-09', 1, '4.99', '2022-02-13', NULL),  (5, 14, 8, '2022-02-23', 2, '6.99', '2022-02-24', NULL),  (6, 3, 1, '2022-02-01', 4, '5.99', '2022-02-02', NULL),  (7, 11, 3, '2022-02-16', 2, '6.00', '2022-02-28', NULL),  (8, 10, 10, '2022-02-09', 1, '4.00', '2022-02-13', NULL); | |
| **Review Table** | |
|  | |
| **SQL Explanation:**  Similar to the Order table the review table makes use of foreign id keys from the customer and book tables. It also has its own primary key in the form of “review id”. Like order id key this is unique to each review and will automatically increment when a review is added to the database.  The other two fields in the review table are the rating and comment field.The table will allow the same book id and customer multiple times as users can write reviews for several books however the review ID is unique to that review  **\*The rating is limited to an integer between 1-5** and the comment(review itself) is limited to 255 characters so that they remain brief and along with the rating quickly get the customers point across. | |
| **Data Definition Language used to create table structure** | |
| DROP TABLE IF EXISTS `reviews`;  CREATE TABLE IF NOT EXISTS `reviews` (  `CustomerID` int(11) NOT NULL,  `BookID` int(4) NOT NULL,  `ReviewID` int(11) NOT NULL AUTO\_INCREMENT,  `Rating` int(11) NOT NULL,  `Comment` varchar(255) COLLATE latin1\_bin NOT NULL,  PRIMARY KEY (`ReviewID`),  KEY `reviews\_books\_fk` (`BookID`),  KEY `reviews\_Cust\_fk` (`CustomerID`)  ) ENGINE=InnoDB AUTO\_INCREMENT=12 DEFAULT CHARSET=latin1 COLLATE=latin1\_bin; | |
| **Data Definition Language used to link table with other tables in database** | |
| ALTER TABLE `reviews`  ADD CONSTRAINT `reviews\_Cust\_fk` FOREIGN KEY (`CustomerID`) REFERENCES `customer` (`Customer\_ID`),  ADD CONSTRAINT `reviews\_books\_fk` FOREIGN KEY (`BookID`) REFERENCES `books` (`BookID`);  COMMIT; | |
| **Data Manipulation Language used to populate table** | |
| INSERT INTO `reviews` (`CustomerID`, `BookID`, `ReviewID`, `Rating`, `Comment`) VALUES  (17, 13, 1, 4, 'Very Good'),  (5, 5, 2, 5, 'Loved it, would recommend to everyone'),  (7, 4, 3, 3, ''),  (15, 3, 4, 3, 'Ok'),  (3, 2, 5, 5, 'Just brilliant'),  (23, 6, 6, 4, 'Enjoyed it'),  (11, 8, 7, 3, 'Ok, but these is better books'),  (16, 7, 8, 2, 'Disappointing'),  (7, 9, 9, 3, 'liked it, just'),  (25, 15, 10, 5, 'Really really good'),  (6, 10, 11, 2, 'Needs work'); | |
| **SQL Queries to be used to create ecommerce functionality** | |
| **SQL Queries to be used in Browse Products webpage - Conor Clyde**  **(Data Query Language)** | |
| SELECT \* FROM Book ORDER BY Book\_ID | |
| SQL Explanation:   * Need SQL Command to retrieve books from the database to display on the webpage * Shown command retrieves each book in the book table and its value for every column, and orders these books by their book ID | |
| **SQL Queries to be used in Individual Product webpage - Conor Clyde**  **(Data Query Language)** | |
| SELECT \* FROM book WHERE Book\_ID = value  SELECT \*  FROM Book  INNER JOIN Stock  ON Book.Book\_ID = Stock.Book\_ID WHERE Book.Book\_ID = value | |
| **SQL Explanation:**   * First Statement: Retrieves each book in the book table with a specified book\_ID value and its value for every column * Second Statement: Retrieves each book in the book table with a specified book\_ID value and its value for every column as well as the related values in the stock table | |
| **SQL Queries to be used in Shopping Cart webpage - Conor Clyde**  **(Data Query Language and Data Manipulation Language)** | |
| SELECT \* FROM book WHERE Book\_ID = value  SELECT \* FROM Book WHERE Book\_ID IN (values)  INSERT INTO `ORDER`  (Customer\_ID)  VALUES (customer\_id\_value)  select \* from `Order` ORDER BY Order\_ID DESC LIMIT 1;  INSERT INTO `ORDER\_BOOK`  (Order\_ID, Book\_ID, Quantity)  VALUES (order\_id\_value, book\_id\_value, quantity\_value)  SELECT \*  FROM Book  INNER JOIN Stock  ON Book.Book\_ID = Stock.Book\_ID WHERE Book.Book\_ID = value | |
| **SQL Explanation:**   * First Statement: Retrieves each book in the book table with a specified book\_ID value and its value for every column * Second Statement: Retrieves each book in the book table where the book\_ID value is equal to any of the multiple values provided and its value for every column * Third Statement:: Insert a new row into the Order Table which fills in the customer\_id column with the particular value that is being passed in * Fourth Statement: Retrieves the last order in the order table and its value for every column * Fifth statement: Inserts a new row into the order\_book table which fills in the order\_id, book\_id, and quantity columns with the particularl values that are being passed in * Sixth Statement: Retrieves each book in the book table with a specified book\_ID value and its value for every column as well as the related values in the stock table | |
| **Book\_Summary View - Shea Conway** | |
|  | |
| **Data Query Language for Book\_Summary** | |
| CREATE VIEW `book\_summary` AS SELECT `book`.`Book\_ID` AS `Book\_ID`, `book`.`Book\_Title` AS `Book\_Title`, `book`.`Book\_Desc` AS `Book\_Desc`, `book`.`Price` AS `Price`, `book`.`Genre` AS `Genre`, `book`.`Book\_Length` AS `Book\_Length` FROM `book` ORDER BY `book`.`Book\_ID` ASC ; | |
| **SQL Explanation:**   * The attributes Book\_ID, Book\_Title, Book\_Desc, Price, Genre and Book\_Length are selected from every record in the table and displayed in ascending order by Book\_ID. This can be used to quickly locate important information about a specific book. | |
| **Cust\_Details View - Shea Conway** | |
|  | |
| **Data Query Language for Cust\_Details** | |
| CREATE VIEW `cust\_details `AS SELECT `customer`.`Customer\_ID` AS `Customer\_ID`, `customer`.`Forename` AS `Forename`, `customer`.`Surname` AS `Surname`, `customer`.`Address` AS `Address`, `customer`.`Postcode` AS `Postcode`, `customer\_management`.`Email` AS `Email`, `customer\_phone\_no`.`Country\_Code` AS `Country\_Code`, `customer\_phone\_no`.`Phone\_No` AS `Phone\_No` FROM ((`customer` join `customer\_management` on((`customer\_management`.`Customer\_ID` = `customer`.`Customer\_ID`))) join `customer\_phone\_no` on((`customer\_phone\_no`.`Customer\_ID` = `customer`.`Customer\_ID`))) ; | |
| **SQL Explanation:**   * The attributes Customer\_ID, Forename, Surname, Address, Postcode, Email, Country\_Code and Phone\_No are selected from the Customer, Customer\_Management, Customer\_Phone\_No tables. * These tables are joined by the Customer\_ID primary key and foreign keys. This view joins each customer table to show all relevant customer details in one view. Customer password is not included for security. | |
| **Low\_Stocklevels View - Shea Conway** | |
|  | |
| **Data Query Language for low\_stocklevels** | |
| CREATE VIEW `low\_stocklevels` AS SELECT `stock`.`Book\_ID` AS `Book\_ID`, `stock`.`book\_Supplier\_ID` AS `supplier\_ID`, `stock`.`Stock\_Level` AS `Stock\_Level`, `supplier`.`supplier\_Name` AS `Supplier\_Name`, `supplier`.`supplier\_Contact\_Name` AS `Supplier\_Contact\_Name`, `supplier`.`contact\_Number` AS `contact\_Number`, `supplier`.`email` AS `email` FROM (`stock` join `supplier`) WHERE ((`stock`.`book\_Supplier\_ID` = `supplier`.`supplier\_ID`) AND (`stock`.`Stock\_Level` < 20)) ORDER BY `stock`.`Stock\_Level` ASC ; | |
| SQL Explanation:   * The attributes Book\_ID, Book\_Supplier\_ID, Stock\_Level, Supplier\_Name, Supplier\_Contact\_Name, Contact\_Number and Email are selected from the Stock and Supplier table. * Records displayed have matching Supplier\_ID in the Stock and Supplier table and have a Stock\_Level which is less than 20. * Records are ordered by Stock\_Level in ascending order. This view will show all the books that have low stock and show relevant supplier contact information. | |
| **Poor\_Reviews View - Shea Conway** | |
|  | |
| **Data Query Language for poor\_reviews** | |
| CREATE VIEW `poor\_reviews` AS SELECT `review`.`Book\_ID` AS `Book\_ID`, `review`.`Review\_ID` AS `Review\_ID`, `review`.`Rating` AS `Rating`, `review`.`Comment` AS `Comment` FROM `review` WHERE (`review`.`Rating` < 3) ; | |
| **SQL Explanation:**   * The attributes Book\_ID, Review\_ID, Rating and Comment are selected from the Review table. * Records which have a rating number which is less than 3 are displayed. This view will show every book and its review that have a bad rating. | |