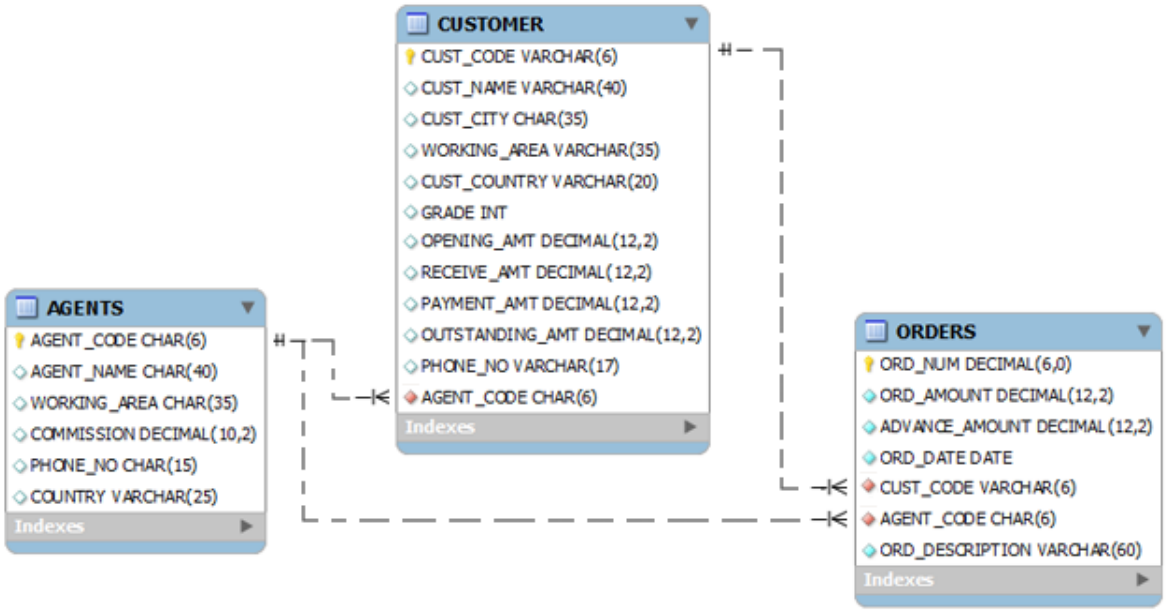
Project 5: Sales and Support Analysis using SQL

Fliipazon Retail is a multinational company that operates a chain of retail stores across different regions. They sell a wide range of products and want to analyse their sales data to gain insights into their business performance and make informed decisions. They have a relational database that stores the following data:



Query for MySQL:

create database interview;

use interview;

```
CREATE TABLE IF NOT EXISTS `agents` (
  `AGENT_CODE` varchar(6) NOT NULL DEFAULT '',
  `AGENT_NAME` varchar(40) DEFAULT NULL,
  `WORKING_AREA` varchar(35) DEFAULT NULL,
  `COMMISSION` decimal(10,2) DEFAULT NULL,
  `PHONE_NO` varchar(15) DEFAULT NULL,
  `COUNTRY` varchar(25) DEFAULT NULL,
  PRIMARY KEY (`AGENT_CODE`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `agents`
--
```

```
INSERT INTO `agents` (`AGENT_CODE`, `AGENT_NAME`, `WORKING_AREA`,
`COMMISSION`, `PHONE_NO`, `COUNTRY`) VALUES
('A007  ', 'Ramasundar                ', 'Bangalore                ', '0.15',
'077-25814763   ', '\r'),
('A003  ', 'Alex                      ', 'London                   ', '0.13',
'075-12458969   ', '\r'),
('A008  ', 'Alford                    ', 'New York                 ', '0.12',
'044-25874365   ', '\r'),
('A011  ', 'Ravi Kumar                ', 'Bangalore                ', '0.15',
'077-45625874   ', '\r'),
('A010  ', 'Santakumar                ', 'Chennai                  ', '0.14',
'007-22388644   ', '\r'),
('A012  ', 'Lucida                    ', 'San Jose                 ', '0.12',
'044-52981425   ', '\r'),
('A005  ', 'Anderson                  ', 'Brisban                  ', '0.13',
'045-21447739   ', '\r'),
('A001  ', 'Subbarao                  ', 'Bangalore                ', '0.14',
'077-12346674   ', '\r'),
('A002  ', 'Mukesh                    ', 'Mumbai                   ', '0.11',
'029-12358964   ', '\r'),
('A006  ', 'McDen                     ', 'London                   ', '0.15',
'078-22255588   ', '\r'),
('A004  ', 'Ivan                      ', 'Torento                  ', '0.15',
'008-22544166   ', '\r'),
('A009  ', 'Benjamin                  ', 'Hampshair                ', '0.11',
'008-22536178   ', '\r');

--
-- Table structure for table `customer`
--

CREATE TABLE IF NOT EXISTS `customer` (
  `CUST_CODE` varchar(6) NOT NULL,
  `CUST_NAME` varchar(40) NOT NULL,
  `CUST_CITY` varchar(35) DEFAULT NULL,
  `WORKING_AREA` varchar(35) NOT NULL,
  `CUST_COUNTRY` varchar(20) NOT NULL,
  `GRADE` decimal(10,0) DEFAULT NULL,
  `OPENING_AMT` decimal(12,2) NOT NULL,
  `RECEIVE_AMT` decimal(12,2) NOT NULL,
  `PAYMENT_AMT` decimal(12,2) NOT NULL,
  `OUTSTANDING_AMT` decimal(12,2) NOT NULL,
  `PHONE_NO` varchar(17) NOT NULL,
  `AGENT_CODE` varchar(6) DEFAULT NULL,
```

```
  KEY `CUSTCITY` (`CUST_CITY`),
  KEY `CUSTCITY_COUNTRY` (`CUST_CITY`,`CUST_COUNTRY`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `customer`
--

INSERT INTO `customer` (`CUST_CODE`, `CUST_NAME`, `CUST_CITY`,
`WORKING_AREA`, `CUST_COUNTRY`, `GRADE`, `OPENING_AMT`,
`RECEIVE_AMT`, `PAYMENT_AMT`, `OUTSTANDING_AMT`, `PHONE_NO`,
`AGENT_CODE`) VALUES
('C00013', 'Holmes', 'London                    ', 'London', 'UK', '2', '6000.00',
'5000.00', '7000.00', '4000.00', 'BBBBBBB', 'A003  '),
('C00001', 'Micheal', 'New York                  ', 'New York', 'USA', '2', '3000.00',
'5000.00', '2000.00', '6000.00', 'CCCCCCC', 'A008  '),
('C00020', 'Albert', 'New York                   ', 'New York', 'USA', '3', '5000.00',
'7000.00', '6000.00', '6000.00', 'BBBBSBB', 'A008  '),
('C00025', 'Ravindran', 'Bangalore               ', 'Bangalore', 'India', '2', '5000.00',
'7000.00', '4000.00', '8000.00', 'AVAVAVA', 'A011  '),
('C00024', 'Cook', 'London                       ', 'London', 'UK', '2', '4000.00', '9000.00',
'7000.00', '6000.00', 'FSDDSDF', 'A006  '),
('C00015', 'Stuart', 'London                     ', 'London', 'UK', '1', '6000.00',
'8000.00', '3000.00', '11000.00', 'GFSGERS', 'A003  '),
('C00002', 'Bolt', 'New York                     ', 'New York', 'USA', '3', '5000.00',
'7000.00', '9000.00', '3000.00', 'DDNRDRH', 'A008  '),
('C00018', 'Fleming', 'Brisban                   ', 'Brisban', 'Australia', '2', '7000.00',
'7000.00', '9000.00', '5000.00', 'NHBGVFC', 'A005  '),
('C00021', 'Jacks', 'Brisban                     ', 'Brisban', 'Australia', '1', '7000.00',
'7000.00', '7000.00', '7000.00', 'WERTGDF', 'A005  '),
('C00019', 'Yearannaidu', 'Chennai               ', 'Chennai', 'India', '1', '8000.00',
'7000.00', '7000.00', '8000.00', 'ZZZZBFV', 'A010  '),
('C00005', 'Sasikant', 'Mumbai                   ', 'Mumbai', 'India', '1', '7000.00',
'11000.00', '7000.00', '11000.00', '147-25896312', 'A002  '),
('C00007', 'Ramanathan', 'Chennai                ', 'Chennai', 'India', '1', '7000.00',
'11000.00', '9000.00', '9000.00', 'GHRDWSD', 'A010  '),
('C00022', 'Avinash', 'Mumbai                    ', 'Mumbai', 'India', '2', '7000.00',
'11000.00', '9000.00', '9000.00', '113-12345678', 'A002  '),
('C00004', 'Winston', 'Brisban                   ', 'Brisban', 'Australia', '1', '5000.00',
'8000.00', '7000.00', '6000.00', 'AAAAAAA', 'A005  '),
('C00023', 'Karl', 'London                       ', 'London', 'UK', '0', '4000.00', '6000.00',
'7000.00', '3000.00', 'AAAABAA', 'A006  '),
('C00006', 'Shilton', 'Torento                   ', 'Torento', 'Canada', '1', '10000.00',
'7000.00', '6000.00', '11000.00', 'DDDDDDD', 'A004  '),
```

```
('C00010', 'Charles', 'Hampshair                    ', 'Hampshair', 'UK', '3', '6000.00',
'4000.00', '5000.00', '5000.00', 'MMMMMMM', 'A009  '),
('C00017', 'Srinivas', 'Bangalore                   ', 'Bangalore', 'India', '2', '8000.00',
'4000.00', '3000.00', '9000.00', 'AAAAAAB', 'A007  '),
('C00012', 'Steven', 'San Jose                      ', 'San Jose', 'USA', '1', '5000.00',
'7000.00', '9000.00', '3000.00', 'KRFYGJK', 'A012  '),
('C00008', 'Karolina', 'Torento                     ', 'Torento', 'Canada', '1', '7000.00',
'7000.00', '9000.00', '5000.00', 'HJKORED', 'A004  '),
('C00003', 'Martin', 'Torento                       ', 'Torento', 'Canada', '2', '8000.00',
'7000.00', '7000.00', '8000.00', 'MJYURFD', 'A004  '),
('C00009', 'Ramesh', 'Mumbai                        ', 'Mumbai', 'India', '3', '8000.00',
'7000.00', '3000.00', '12000.00', 'Phone No', 'A002  '),
('C00014', 'Rangarappa', 'Bangalore                 ', 'Bangalore', 'India', '2',
'8000.00', '11000.00', '7000.00', '12000.00', 'AAAATGF', 'A001  '),
('C00016', 'Venkatpati', 'Bangalore                 ', 'Bangalore', 'India', '2', '8000.00',
'11000.00', '7000.00', '12000.00', 'JRTVFDD', 'A007  '),
('C00011', 'Sundariya', 'Chennai                    ', 'Chennai', 'India', '3', '7000.00',
'11000.00', '7000.00', '11000.00', 'PPHGRTS', 'A010  ');

-- Table structure for table `orders`
--

CREATE TABLE IF NOT EXISTS `orders` (
  `ORD_NUM` decimal(6,0) NOT NULL,
  `ORD_AMOUNT` decimal(12,2) NOT NULL,
  `ADVANCE_AMOUNT` decimal(12,2) NOT NULL,
  `ORD_DATE` date NOT NULL,
  `CUST_CODE` varchar(6) NOT NULL,
  `AGENT_CODE` varchar(6) NOT NULL,
  `ORD_DESCRIPTION` varchar(60) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `orders`
--

INSERT INTO `orders` (`ORD_NUM`, `ORD_AMOUNT`, `ADVANCE_AMOUNT`,
`ORD_DATE`, `CUST_CODE`, `AGENT_CODE`, `ORD_DESCRIPTION`) VALUES
('200100', '1000.00', '600.00', '2008-01-08', 'C00015', 'A003  ', 'SOD\r'),
('200110', '3000.00', '500.00', '2008-04-15', 'C00019', 'A010  ', 'SOD\r'),
('200107', '4500.00', '900.00', '2008-08-30', 'C00007', 'A010  ', 'SOD\r'),
('200112', '2000.00', '400.00', '2008-05-30', 'C00016', 'A007  ', 'SOD\r'),
('200113', '4000.00', '600.00', '2008-06-10', 'C00022', 'A002  ', 'SOD\r'),
('200102', '2000.00', '300.00', '2008-05-25', 'C00012', 'A012  ', 'SOD\r'),
```

('200114', '3500.00', '2000.00', '2008-08-15', 'C00002', 'A008 ', 'SOD\r'),
('200122', '2500.00', '400.00', '2008-09-16', 'C00003', 'A004 ', 'SOD\r'),
('200118', '500.00', '100.00', '2008-07-20', 'C00023', 'A006 ', 'SOD\r'),
('200119', '4000.00', '700.00', '2008-09-16', 'C00007', 'A010 ', 'SOD\r'),
('200121', '1500.00', '600.00', '2008-09-23', 'C00008', 'A004 ', 'SOD\r'),
('200130', '2500.00', '400.00', '2008-07-30', 'C00025', 'A011 ', 'SOD\r'),
('200134', '4200.00', '1800.00', '2008-09-25', 'C00004', 'A005 ', 'SOD\r'),
('200115', '2000.00', '1200.00', '2008-02-08', 'C00013', 'A013 ', 'SOD\r'),
('200108', '4000.00', '600.00', '2008-02-15', 'C00008', 'A004 ', 'SOD\r'),
('200103', '1500.00', '700.00', '2008-05-15', 'C00021', 'A005 ', 'SOD\r'),
('200105', '2500.00', '500.00', '2008-07-18', 'C00025', 'A011 ', 'SOD\r'),
('200109', '3500.00', '800.00', '2008-07-30', 'C00011', 'A010 ', 'SOD\r'),
('200101', '3000.00', '1000.00', '2008-07-15', 'C00001', 'A008 ', 'SOD\r'),
('200111', '1000.00', '300.00', '2008-07-10', 'C00020', 'A008 ', 'SOD\r'),
('200104', '1500.00', '500.00', '2008-03-13', 'C00006', 'A004 ', 'SOD\r'),
('200106', '2500.00', '700.00', '2008-04-20', 'C00005', 'A002 ', 'SOD\r'),
('200125', '2000.00', '600.00', '2008-10-10', 'C00018', 'A005 ', 'SOD\r'),
('200117', '800.00', '200.00', '2008-10-20', 'C00014', 'A001 ', 'SOD\r'),
('200123', '500.00', '100.00', '2008-09-16', 'C00022', 'A002 ', 'SOD\r'),
('200120', '500.00', '100.00', '2008-07-20', 'C00009', 'A002 ', 'SOD\r'),
('200116', '500.00', '100.00', '2008-07-13', 'C00010', 'A009 ', 'SOD\r'),
('200124', '500.00', '100.00', '2008-06-20', 'C00017', 'A007 ', 'SOD\r'),
('200126', '500.00', '100.00', '2008-06-24', 'C00022', 'A002 ', 'SOD\r'),
('200129', '2500.00', '500.00', '2008-07-20', 'C00024', 'A006 ', 'SOD\r'),
('200127', '2500.00', '400.00', '2008-07-20', 'C00015', 'A003 ', 'SOD\r'),
('200128', '3500.00', '1500.00', '2008-07-20', 'C00009', 'A002 ', 'SOD\r'),
('200135', '2000.00', '800.00', '2008-09-16', 'C00007', 'A010 ', 'SOD\r'),
('200131', '900.00', '150.00', '2008-08-26', 'C00012', 'A012 ', 'SOD\r'),
('200133', '1200.00', '400.00', '2008-06-29', 'C00009', 'A002 ', 'SOD\r'),
('200132', '4000.00', '2000.00', '2008-08-15', 'C00013', 'A013 ', 'SOD\r');

Objective:
The objective of this project is to analyse this data using SQL queries and generate meaningful reports to support decision-making processes. The reports should provide insights into sales trends, customer behaviour and support agent performance.

To achieve so, they have the following business objectives:

1. Establish Chinook as the premier destination for music records.
2. Provide a diverse selection of music genres to cater to a wide range of customer preferences.
3. Deliver excellent customer service to enhance the shopping experience.

4. Build a loyal customer base and foster long-term relationships through engagement and promotional events.

To drive its decisions through data, Chinook maintains a comprehensive database capturing various aspects of its business. The database includes information about different albums, music tracks, customers, sales transactions, and employees. Your task is to utilise your SQL skills to conduct data cleaning and analysis, enabling the extraction of valuable insights that will drive business growth and ensure long-term success.

Segment 1: Database - Tables, Columns, Relationships

- Identify the tables in the database and their respective columns.
- Determine the number of records in each table within the schema.
- Identify and handle any missing or inconsistent values in the dataset.
- Analyse the data types of the columns in each table to ensure they are appropriate for the stored data.
- Identify any duplicate records within the tables and develop a strategy for handling them.

Segment 2: Basic Sales Analysis

- Write SQL queries to retrieve the total number of orders, total revenue, and average order value.
- The operations team needs to track the agent who has handled the maximum number of high-grade customers. Write a SQL query to find the agent_name who has the highest count of customers with a grade of 5. Display the agent_name and the count of high-grade customers.
- The company wants to identify the most active customer cities in terms of the total order amount. Write a SQL query to find the top 3 customer cities with the highest total order amount. Include cust_city and total_order_amount in the output.

Segment 3: Customer Analysis:

- Calculate the total number of customers.
- Identify the top-spending customers based on their total order value.
- Analyse customer retention by calculating the percentage of repeat customers.
- Find the name of the customer who has the maximum outstanding amount from every country.
- Write a SQL query to calculate the percentage of customers in each grade category (1 to 5).

Segment 4: Agent Performance Analysis

- Company wants to provide a performance bonus to their best agents based on the maximum order amount. Find the top 5 agents eligible for it.
- The company wants to analyse the performance of agents based on the number of orders they have handled. Write a SQL query to rank agents based on the total number of orders they have processed. Display agent_name, total_orders, and their respective ranking.

- Company wants to change the commission for the agents, basis advance payment they collected. Write a sql query which creates a new column updated_commision on the basis below rules.
    - If the average advance amount collected is less than 750, there is no change in commission.
    - If the average advance amount collected is between 750 and 1000 (inclusive), the new commission will be 1.5 times the old commission.
    - If the average advance amount collected is more than 1000, the new commission will be 2 times the old commission.

Segment 5: SQL Tasks

- Add a new column named avg_rcv_amt in the table customers which contains the average receive amount for every country. Display all columns from the customer table along with the avg_rcv_amt column in the last.
- Write a sql query to create and call a UDF named avg_amt to return the average outstanding amount of the customers which are managed by a given agent. Also, call the UDF with the agent name ‘Mukesh’.
- Write a sql query to create and call a subroutine called cust_detail to return all the details of the customer which are having the given grade. Also, call the subroutine with grade 2.
- Write a stored procedure sp_name which will return the concatenated ord_num (comma separated) of the customer with input customer code using cursor. Also, write the procedure call query with cust_code ‘C00015’.

Evaluation pointers:

- The tasks are correctly identified and executed.
- The solution output matches the expected output.
- The query is optimised and syntactically correct.
- Proper aliases are used
- If required any, appropriate comments are written.
- The code is written concisely with appropriate indentations.