Partie 1 -

1. Création de la BDD et des tableaux.

CREATE TABLE categories (

codeCate VARCHAR(10) PRIMARY KEY,

libelle VARCHAR(50)

);

CREATE TABLE grilleTarifs (

codeGam VARCHAR(10),

codeCate VARCHAR(10),

codeTarif VARCHAR(10),

PRIMARY KEY (codeGam, codeCate, codeTarif),

FOREIGN KEY (codeGam) REFERENCES gammes(codeGam),

FOREIGN KEY (codeCate) REFERENCES categories(codeCate),

FOREIGN KEY (codeTarif) REFERENCES tarifs(codeTarif)

);

CREATE TABLE articles (

refart VARCHAR(10) PRIMARY KEY,

designation VARCHAR(100),

codeGam VARCHAR(10),

codeCate VARCHAR(10),

FOREIGN KEY (codeGam) REFERENCES gammes(codeGam),

FOREIGN KEY (codeCate) REFERENCES categories(codeCate)

);

CREATE TABLE lignesFic (

noFic INT,

noLig INT,

refart VARCHAR(10),

depart DATE,

retour DATE NULL,

PRIMARY KEY (noFic, noLig),

FOREIGN KEY (noFic) REFERENCES fiches(noFic),

FOREIGN KEY (refart) REFERENCES articles(refart)

);

CREATE DATABASE IF NOT EXISTS location\_ski;

USE location\_ski;

CREATE TABLE clients (

noCli INT PRIMARY KEY,

nom VARCHAR(50),

prenom VARCHAR(50),

adresse VARCHAR(100),

cpo VARCHAR(10),

ville VARCHAR(50)

);

CREATE TABLE fiches (

noFic INT PRIMARY KEY,

noCli INT,

dateCrea DATE,

datePaiement DATE NULL,

etat ENUM('SO', 'EC', 'RE'),

FOREIGN KEY (noCli) REFERENCES clients(noCli)

);

CREATE TABLE tarifs (

codeTarif VARCHAR(10) PRIMARY KEY,

libelle VARCHAR(50),

prixJour DECIMAL(5,2)

);

CREATE TABLE gammes (

codeGam VARCHAR(10) PRIMARY KEY,

libelle VARCHAR(50)

);

1. Insertion des données :

USE location\_ski;

INSERT INTO clients (noCli, nom, prenom, adresse, cpo, ville) VALUES

(1, 'Albert', 'Anatole', 'Rue des accacias', '61000', 'Amiens'),

(2, 'Bernard', 'Barnabé', 'Rue du bar', '1000', 'Bourg en Bresse'),

(3, 'Dupond', 'Camille', 'Rue Crébillon', '44000', 'Nantes'),

(4, 'Desmoulin', 'Daniel', 'Rue descendante', '21000', 'Dijon'),

(5, 'Ernest', 'Etienne', 'Rue de l’échaffaud', '42000', 'Saint Étienne'),

(6, 'Ferdinand', 'François', 'Rue de la convention', '44100', 'Nantes'),

(9, 'Dupond', 'Jean', 'Rue des mimosas', '75018', 'Paris'),

(14, 'Boutaud', 'Sabine', 'Rue des platanes', '75002', 'Paris');

INSERT INTO fiches (noFic, noCli, dateCrea, datePaiement, etat) VALUES

(1001, 14, DATE\_SUB(NOW(),INTERVAL 15 DAY), DATE\_SUB(NOW(),INTERVAL 13 DAY),'SO' ),

(1002, 4, DATE\_SUB(NOW(),INTERVAL 13 DAY), NULL, 'EC'),

(1003, 1, DATE\_SUB(NOW(),INTERVAL 12 DAY), DATE\_SUB(NOW(),INTERVAL 10 DAY),'SO'),

(1004, 6, DATE\_SUB(NOW(),INTERVAL 11 DAY), NULL, 'EC'),

(1005, 3, DATE\_SUB(NOW(),INTERVAL 10 DAY), NULL, 'EC'),

(1006, 9, DATE\_SUB(NOW(),INTERVAL 10 DAY),NULL ,'RE'),

(1007, 1, DATE\_SUB(NOW(),INTERVAL 3 DAY), NULL, 'EC'),

(1008, 2, DATE\_SUB(NOW(),INTERVAL 0 DAY), NULL, 'EC');

INSERT INTO tarifs (codeTarif, libelle, prixJour) VALUES

('T1', 'Base', 10),

('T2', 'Chocolat', 15),

('T3', 'Bronze', 20),

('T4', 'Argent', 30),

('T5', 'Or', 50),

('T6', 'Platine', 90);

INSERT INTO gammes (codeGam, libelle) VALUES

('PR', 'Matériel Professionnel'),

('HG', 'Haut de gamme'),

('MG', 'Moyenne gamme'),

('EG', 'Entrée de gamme');

INSERT INTO categories (codeCate, libelle) VALUES

('MONO', 'Monoski'),

('SURF', 'Surf'),

('PA', 'Patinette'),

('FOA', 'Ski de fond alternatif'),

('FOP', 'Ski de fond patineur'),

('SA', 'Ski alpin');

INSERT INTO grilleTarifs (codeGam, codeCate, codeTarif) VALUES

('EG', 'MONO', 'T1'),

('MG', 'MONO', 'T2'),

('EG', 'SURF', 'T1'),

('MG', 'SURF', 'T2'),

('HG', 'SURF', 'T3'),

('PR', 'SURF', 'T5'),

('EG', 'PA', 'T1'),

('MG', 'PA', 'T2'),

('EG', 'FOA', 'T1'),

('MG', 'FOA', 'T2'),

('HG', 'FOA', 'T4'),

('PR', 'FOA', 'T6'),

('EG', 'FOP', 'T2'),

('MG', 'FOP', 'T3'),

('HG', 'FOP', 'T4'),

('PR', 'FOP', 'T6'),

('EG', 'SA', 'T1'),

('MG', 'SA', 'T2'),

('HG', 'SA', 'T4'),

('PR', 'SA', 'T6');

INSERT INTO articles (refart, designation, codeGam, codeCate) VALUES

('F01', 'Fischer Cruiser', 'EG', 'FOA'),

('F02', 'Fischer Cruiser', 'EG', 'FOA'),

('F03', 'Fischer Cruiser', 'EG', 'FOA'),

('F04', 'Fischer Cruiser', 'EG', 'FOA'),

('F05', 'Fischer Cruiser', 'EG', 'FOA'),

('F10', 'Fischer Sporty Crown', 'MG', 'FOA'),

('F20', 'Fischer RCS Classic GOLD', 'PR', 'FOA'),

('F21', 'Fischer RCS Classic GOLD', 'PR', 'FOA'),

('F22', 'Fischer RCS Classic GOLD', 'PR', 'FOA'),

('F23', 'Fischer RCS Classic GOLD', 'PR', 'FOA'),

('F50', 'Fischer SOSSkating VASA', 'HG', 'FOP'),

('F60', 'Fischer RCS CARBOLITE Skating', 'PR', 'FOP'),

('F61', 'Fischer RCS CARBOLITE Skating', 'PR', 'FOP'),

('F62', 'Fischer RCS CARBOLITE Skating', 'PR', 'FOP'),

('F63', 'Fischer RCS CARBOLITE Skating', 'PR', 'FOP'),

('F64', 'Fischer RCS CARBOLITE Skating', 'PR', 'FOP'),

('P01', 'Décathlon Allegre junior 150', 'EG', 'PA'),

('P10', 'Fischer mini ski patinette', 'MG', 'PA'),

('P11', 'Fischer mini ski patinette', 'MG', 'PA'),

('S01', 'Décathlon Apparition', 'EG', 'SURF'),

('S02', 'Décathlon Apparition', 'EG', 'SURF'),

('S03', 'Décathlon Apparition', 'EG', 'SURF'),

('A10', 'Salomon Pro Link Equipe 4S', 'PR', 'SA'),

('A11', 'Salomon Pro Link Equipe 4S', 'PR', 'SA'),

('A21', 'Salomon 3V RACE JR+L10', 'PR', 'SA');

('A01', 'Salomon 24X+Z12', 'EG', 'SA'),

('A02', 'Salomon 24X+Z12', 'EG', 'SA'),

('A03', 'Salomon 24X+Z12', 'EG', 'SA'),

('A04', 'Salomon 24X+Z12', 'EG', 'SA'),

('A05', 'Salomon 24X+Z12', 'EG', 'SA'),

INSERT INTO lignesFic (noFic, noLig, refart, depart, retour) VALUES

(1001, 1, 'F05', DATE\_SUB(NOW(),INTERVAL 15 DAY), DATE\_SUB(NOW(),INTERVAL 13 DAY)),

(1001, 2, 'F50', DATE\_SUB(NOW(),INTERVAL 15 DAY), DATE\_SUB(NOW(),INTERVAL 14 DAY)),

(1001, 3, 'F60', DATE\_SUB(NOW(),INTERVAL 13 DAY), DATE\_SUB(NOW(),INTERVAL 13 DAY)),

(1002, 1, 'A03', DATE\_SUB(NOW(),INTERVAL 13 DAY), DATE\_SUB(NOW(),INTERVAL 9 DAY)),

(1002, 2, 'A04', DATE\_SUB(NOW(),INTERVAL 12 DAY), DATE\_SUB(NOW(),INTERVAL 7 DAY)),

(1002, 3, 'S03', DATE\_SUB(NOW(),INTERVAL 8 DAY), NULL),

(1003, 1, 'F50', DATE\_SUB(NOW(),INTERVAL 12 DAY), DATE\_SUB(NOW(),INTERVAL 10 DAY)),

(1003, 2, 'F05', DATE\_SUB(NOW(),INTERVAL 12 DAY), DATE\_SUB(NOW(),INTERVAL 10 DAY)),

(1004, 1, 'P01', DATE\_SUB(NOW(),INTERVAL 6 DAY), NULL),

(1005, 1, 'F05', DATE\_SUB(NOW(),INTERVAL 9 DAY), DATE\_SUB(NOW(),INTERVAL 5 DAY)),

(1005, 2, 'F10', DATE\_SUB(NOW(),INTERVAL 4 DAY), NULL),

(1006, 1, 'S01', DATE\_SUB(NOW(),INTERVAL 10 DAY), DATE\_SUB(NOW(),INTERVAL 9 DAY)),

(1006, 2, 'S02', DATE\_SUB(NOW(),INTERVAL 10 DAY), DATE\_SUB(NOW(),INTERVAL 9 DAY)),

(1006, 3, 'S03', DATE\_SUB(NOW(),INTERVAL 10 DAY), DATE\_SUB(NOW(),INTERVAL 9 DAY)),

(1007, 1, 'F50', DATE\_SUB(NOW(),INTERVAL 3 DAY), DATE\_SUB(NOW(),INTERVAL 2 DAY)),

(1007, 3, 'F60', DATE\_SUB(NOW(),INTERVAL 1 DAY), NULL),

(1007, 2, 'F05', DATE\_SUB(NOW(),INTERVAL 3 DAY), NULL),

(1007, 4, 'S02', DATE\_SUB(NOW(),INTERVAL 0 DAY), NULL),

(1008, 1, 'S01', DATE\_SUB(NOW(),INTERVAL 0 DAY), NULL);

Partie 2-

1. Une image contenant texte, capture d’écran, Police, ligne

   Description générée automatiquementListe des clients (toutes les informations) dont le nom commence par un D

SELECT \* FROM clients WHERE nom LIKE 'D%';

1. Une image contenant texte, capture d’écran, Police, nombre

   Description générée automatiquementNom et prénom de tous les clients

SELECT prenom, nom FROM clients;

1. Liste des fiches (n°, état) pour les clients (nom, prénom) qui habitent en Loire Atlantique (44)

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquement

SELECT fiches.noFic, fiches.etat, clients.nom, clients.prenom

FROM fiches JOIN clients ON fiches.noCli = clients.noCli WHERE clients.cpo LIKE '44%';

1. Détail de la fiche n°1002

SELECT fiches.noFic, clients.nom, clients.prenom, lignesFic.refart, articles.designation, lignesFic.depart, lignesFic.retour, tarifs.prixJour,

DATEDIFF(IFNULL(lignesFic.retour, NOW()), lignesFic.depart) \* tarifs.prixJour AS montant

FROM fiches

JOIN clients ON fiches.noCli = clients.noCli

JOIN lignesFic ON fiches.noFic = lignesFic.noFic

JOIN articles ON lignesFic.refart = articles.refart

JOIN grilleTarifs ON articles.codeGam = grilleTarifs.codeGam

AND articles.codeCate = grilleTarifs.codeCate

JOIN tarifs ON grilleTarifs.codeTarif = tarifs.codeTarif

WHERE fiches.noFic = 1002;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. Prix journalier moyen de location par gamme

SELECT gammes.libelle AS Gamme, AVG(tarifs.prixJour) AS tarif\_journalier\_moyen

FROM grilleTarifs

JOIN gammes ON grilleTarifs.codeGam = gammes.codeGam JOIN tarifs ON grilleTarifs.codeTarif = tarifs.codeTarif

GROUP BY gammes.libelle;

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

1. Détail de la fiche n°1002 avec le total

SELECT fiches.noFic, clients.nom, clients.prenom, lignesFic.refart, articles.designation, lignesFic.depart, lignesFic.retour, tarifs.prixJour, DATEDIFF(IFNULL(lignesFic.retour, NOW()), lignesFic.depart) \* tarifs.prixJour AS montant, (SELECT SUM(DATEDIFF(IFNULL(l.retour, NOW()), l.depart) \* t.prixJour)

FROM lignesFic l

JOIN articles a ON l.refart = a.refart

JOIN grilleTarifs g ON a.codeGam = g.codeGam

AND a.codeCate = g.codeCate

JOIN tarifs t ON g.codeTarif = t.codeTarif

WHERE l.noFic = 1002) AS Total

FROM fiches

JOIN clients ON fiches.noCli = clients.noCli

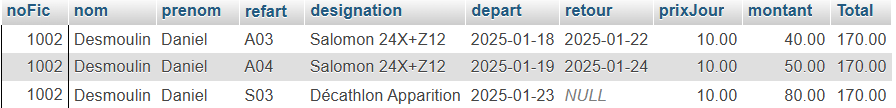
JOIN lignesFic ON fiches.noFic = lignesFic.noFic

JOIN articles ON lignesFic.refart = articles.refart

JOIN grilleTarifs ON articles.codeGam = grilleTarifs.codeGam AND articles.codeCate = grilleTarifs.codeCate

JOIN tarifs ON grilleTarifs.codeTarif = tarifs.codeTarif

WHERE fiches.noFic = 1002;



1. Une image contenant texte, capture d’écran, Police, nombre

   Description générée automatiquementGrille des tarifs

SELECT categories.libelle AS libelle, gammes.libelle AS Gamme, tarifs.libelle AS Tarif, tarifs.prixJour

FROM grilleTarifs

JOIN categories ON grilleTarifs.codeCate = categories.codeCate

JOIN gammes ON grilleTarifs.codeGam = gammes.codeGam

JOIN tarifs ON grilleTarifs.codeTarif = tarifs.codeTarif

ORDER BY categories.libelle, gammes.libelle;

1. Liste des locations de la catégorie SURF

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

SELECT articles.refart, articles.designation,

COUNT(lignesFic.refart) AS nbEmplacement

FROM articles

JOIN lignesFic ON articles.refart = lignesFic.refart

WHERE articles.codeCate = 'SURF'

GROUP BY articles.refart, articles.designation;

1. Calcul du nombre moyen d’articles loués par fiche de location

SELECT AVG(ligne\_count) AS nb\_lignes\_moyen\_par\_fiche

FROM (SELECT COUNT(\*) AS ligne\_count FROM lignesFic GROUP BY noFic) AS subquery;

Une image contenant texte, Police, blanc, capture d’écran

Description générée automatiquement

1. Une image contenant texte, capture d’écran, Police, ligne

   Description générée automatiquementCalcul du nombre de fiches de location établies pour les catégories de location Ski alpin, Surf et Patinette

SELECT categories.libelle AS catégorie, COUNT(lignesFic.noFic) AS numero\_de\_localisation

FROM lignesFic

JOIN articles ON lignesFic.refart = articles.refart

JOIN categories ON articles.codeCate = categories.codeCate

WHERE categories.libelle IN ('Ski alpin', 'Surf', 'Patinette')

GROUP BY categories.libelle;

1. Calcul du montant moyen des fiches de location

SELECT AVG(total) AS montant\_moyen\_d\_une\_fiche\_de\_location

FROM ( SELECT fiches.noFic, SUM(DATEDIFF(IFNULL(lignesFic.retour, NOW()), lignesFic.depart) \* tarifs.prixJour) AS total

FROM fiches

JOIN lignesFic ON fiches.noFic = lignesFic.noFic

JOIN articles ON lignesFic.refart = articles.refart

JOIN grilleTarifs ON articles.codeGam = grilleTarifs.codeGam AND articles.codeCate = grilleTarifs.codeCate

JOIN tarifs ON grilleTarifs.codeTarif = tarifs.codeTarif

GROUP BY fiches.noFic ) AS sous\_requete;

Une image contenant texte, Police, capture d’écran, typographie

Description générée automatiquement