

HEXaware SQL Case Study

Name – Thilak Raaj

Topic - PetPals, The Pet Adoption Platform

1. Provide a SQL script that initializes the database for the Pet Adoption Platform "PetPals".

The screenshot shows the MySQL Workbench interface with a query editor window titled 'Query 1'. The code in the editor is as follows:

```
1 • CREATE DATABASE PetPals;
2 • USE PetPals;
3
4 • ○ CREATE TABLE Pets (
5     PetID INT AUTO_INCREMENT PRIMARY KEY,
6     Name VARCHAR(255) NOT NULL,
7     Age INT,
8     Breed VARCHAR(255) NOT NULL,
9     Type VARCHAR(255) NOT NULL,
10    AvailableForAdoption BIT NOT NULL
11 );
12 • desc table Donations;
13
14
15 • ○ CREATE TABLE Shelters (
16     ShelterID INT AUTO_INCREMENT PRIMARY KEY,
17     Name VARCHAR(255) NOT NULL,
18     Location VARCHAR(255) NOT NULL
19 );
20
21
```

Below the editor, the 'Output' pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
117	12:15:13	desc table AdoptionEvents	1 row(s) returned	0.000 sec / 0.000 sec
118	12:15:31	desc table Participants	1 row(s) returned	0.000 sec / 0.000 sec
119	12:16:54	Show tables	7 row(s) returned	0.000 sec / 0.000 sec
120	12:17:38	drop database PetPals	7 row(s) affected	0.110 sec
121	12:17:49	CREATE DATABASE PetPals	1 row(s) affected	0.015 sec
122	12:17:52	USE PetPals	0 row(s) affected	0.000 sec

2. Create tables for pets, shelters, donations, adoption events, and participants.

The screenshot shows the MySQL Workbench interface with a query editor window titled 'Query 1'. The code in the editor is as follows:

```
40
41 • ○ CREATE TABLE Participants (
42     ParticipantID INT AUTO_INCREMENT PRIMARY KEY,
43     ParticipantName VARCHAR(255) NOT NULL,
44     ParticipantType VARCHAR(255) NOT NULL,
45     EventID INT,
46     FOREIGN KEY (EventID) REFERENCES AdoptionEvents(EventID)
47     ON DELETE SET NULL
48     ON UPDATE SET NULL
49 );
50 • desc table Participants;
51
52 • Show tables;
53
54
55 • INSERT INTO Pets (PetID, Name, Age, Breed, Type, AvailableForAdoption) VALUES
56     (1, 'Buddy', 3, 'Labrador Retriever', 'Dog', 1),
57     (2, 'Whiskers', 2, 'Siamese', 'Cat', 1),
58     (3, 'Max', 5, 'Golden Retriever', 'Dog', 0),
59     (4, 'Snowball', 4, 'Persian', 'Cat', 1),
```

Below the editor, the 'Result Grid' pane shows the list of tables in the PetPals database:

Tables_in_petpals
adoptelets
adoptionevents
donations
owners
participants
pets
shelters

3. Define appropriate primary keys, foreign keys, and constraints.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Query 1

```
22 • CREATE TABLE Donations (
23     DonationID INT AUTO_INCREMENT PRIMARY KEY,
24     DonorName VARCHAR(255) NOT NULL,
25     DonationType VARCHAR(255) NOT NULL,
26     DonationAmount DECIMAL(10, 2),
27     DonationItem VARCHAR(255),
28     DonationDate DATETIME NOT NULL
29 );
30
31
32 • CREATE TABLE AdoptionEvents (
33     EventID INT AUTO_INCREMENT PRIMARY KEY,
34     Eventname VARCHAR(255) NOT NULL,
35     EventDate DATETIME NOT NULL,
36     Location VARCHAR(255) NOT NULL
37 );
38 • desc table AdoptionEvents;
39
40
41 • CREATE TABLE Participants (
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	AdoptionEvents	NULL	ALL	NULL	NULL	NULL	NULL	5	100.00	NULL

Result 53 x

34°C Sunny

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Query 1

```
7     Age INT,
8     Breed VARCHAR(255) NOT NULL,
9     Type VARCHAR(255) NOT NULL,
10    AvailableForAdoption BIT NOT NULL
11 );
12 • desc table Donations;
13
14
15 • CREATE TABLE Shelters (
16     ShelterID INT AUTO_INCREMENT PRIMARY KEY,
17     Name VARCHAR(255) NOT NULL,
18     Location VARCHAR(255) NOT NULL
19 );
20
21
22 • CREATE TABLE Donations (
23     DonationID INT AUTO_INCREMENT PRIMARY KEY,
24     DonorName VARCHAR(255) NOT NULL,
25     DonationType VARCHAR(255) NOT NULL,
26     DonationAmount DECIMAL(10, 2),
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	Donations	NULL	ALL	NULL	NULL	NULL	NULL	10	100.00	NULL

Result 52 x

34°C Sunny

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Query 1

```
1 • CREATE DATABASE PetPals;
2 • USE PetPals;
3
4 • CREATE TABLE Pets (
5     PETID INT AUTO_INCREMENT PRIMARY KEY,
6     Name VARCHAR(255) NOT NULL,
7     Age INT,
8     Breed VARCHAR(255) NOT NULL,
9     Type VARCHAR(255) NOT NULL,
10    AvailableForAdoption BIT NOT NULL
11 );
12 • desc table Shelters;
13
14
15 • CREATE TABLE Shelters (
16     ShelterID INT AUTO_INCREMENT PRIMARY KEY,
17     Name VARCHAR(255) NOT NULL,
18     Location VARCHAR(255) NOT NULL
19 );
20
```

Result Grid | Filter Rows | Export: | Wrap Cell Content: |

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	Shelters	ALL	ALL	NULL	NULL	NULL	NULL	5	100.00	NULL

Result 51 ×

34°C Sunny

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Query 1

```
1 • CREATE DATABASE PetPals;
2 • USE PetPals;
3
4 • CREATE TABLE Pets (
5     PETID INT AUTO_INCREMENT PRIMARY KEY,
6     Name VARCHAR(255) NOT NULL,
7     Age INT,
8     Breed VARCHAR(255) NOT NULL,
9     Type VARCHAR(255) NOT NULL,
10    AvailableForAdoption BIT NOT NULL
11 );
12 • desc table Pets;
13
14
15 • CREATE TABLE Shelters (
16     ShelterID INT AUTO_INCREMENT PRIMARY KEY,
17     Name VARCHAR(255) NOT NULL,
18     Location VARCHAR(255) NOT NULL
19 );
20
```

Result Grid | Filter Rows | Export: | Wrap Cell Content: |

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	Pets	ALL	ALL	NULL	NULL	NULL	NULL	10	100.00	NULL

Result 50 ×

34°C Sunny

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Query 1

```
1 • CREATE DATABASE PetPals;
2 • USE PetPals;
3
4 • CREATE TABLE Pets (
5     PETID INT AUTO_INCREMENT PRIMARY KEY,
6     Name VARCHAR(255) NOT NULL,
7     Age INT,
8     Breed VARCHAR(255) NOT NULL,
9     Type VARCHAR(255) NOT NULL,
10    AvailableForAdoption BIT NOT NULL
11 );
12 • desc table Pets;
13
14
15 • CREATE TABLE Shelters (
16     ShelterID INT AUTO_INCREMENT PRIMARY KEY,
17     Name VARCHAR(255) NOT NULL,
18     Location VARCHAR(255) NOT NULL
19 );
20
```

Result Grid | Filter Rows | Export: | Wrap Cell Content: |

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	Pets	ALL	ALL	NULL	NULL	NULL	NULL	10	100.00	NULL

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Query 1

```

34     Eventname VARCHAR(255) NOT NULL,
35     EventDate DATETIME NOT NULL,
36     Location VARCHAR(255) NOT NULL
37   );
38 •  desc table AdoptionEvents;
39
40
41 • CREATE TABLE Participants (
42     ParticipantID INT AUTO_INCREMENT PRIMARY KEY,
43     ParticipantName VARCHAR(255) NOT NULL,
44     ParticipantType VARCHAR(255) NOT NULL,
45     EventID INT,
46     FOREIGN KEY (EventID) REFERENCES AdoptionEvents(EventID)
47     ON DELETE SET NULL
48     ON UPDATE SET NULL
49   );
50 •  desc table Participants;
51
52
53

```

Result Grid | Filter Rows: Export: Wrap Cell Content:

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	Participants		ALL					10	100.00	

Result 54 x Read Only

34°C Sunny 18:04 2024

4. Ensure the script handles potential errors, such as if the database or tables already exist.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Query 1

```

108     ($, 'Pet Paradise Adoption Event', '2024-04-30 14:00:00', 'Ooty');
109
110
111 •  INSERT INTO Participants (ParticipantID, ParticipantName, ParticipantType, EventID) VALUES
112     (1, 'Happy', 'Shelter', 1),
113     (2, 'Glory', 'Shelter', 2),
114     (3, 'Joy', 'Shelter', 3),
115     (4, 'Fluffy', 'Shelter', 4),
116     (5, 'Cute', 'Shelter', 5),
117     (6, 'Aditya', 'Adopter', 1),
118     (7, 'Riya', 'Adopter', 2),
119     (8, 'Ananya', 'Adopter', 3),
120     (9, 'Vivek', 'Donor', 4),
121     (10, 'Neha', 'Donor', 5);
122
123
124 •  CREATE DATABASE IF NOT EXISTS PetPals;
125
126 •  SELECT IF( (SELECT COUNT(*) FROM INFORMATION_SCHEMA.SCHEMATA WHERE SCHEMA_NAME = 'PetPals') > 0, 'PetPals database already exists.', 'PetPals database created successfully.') AS Message;
127

```

Result Grid | Filter Rows: Export: Wrap Cell Content:

Message
PetPals database already exists.

Result 47 x Read Only

34°C Sunny 18:04 2024

5. Write an SQL query that retrieves a list of available pets (those marked as available for adoption) from the "Pets" table. Include the pet's name, age, breed, and type in the result set. Ensure that the query filters out pets that are not available for adoption.

MySQL Workbench

case

File Edit View Query Database Server Tools Scripting Help

Query 1

Limit to 1000 rows

```
111 • INSERT INTO Participants (ParticipantID, ParticipantName, ParticipantType, EventID) VALUES
112     (1, 'Happy', 'Shelter', 1),
113     (2, 'Glory', 'Shelter', 2),
114     (3, 'Joy', 'Shelter', 3),
115     (4, 'Fluffy', 'Shelter', 4),
116     (5, 'Cute', 'Shelter', 5),
117     (6, 'Aditya', 'Adopter', 1),
118     (7, 'Riya', 'Adopter', 2),
119     (8, 'Ananya', 'Adopter', 3),
120     (9, 'Vivek', 'Donor', 4),
121     (10, 'Neha', 'Donor', 5);
122
123
124 • CREATE DATABASE IF NOT EXISTS PetPals;
125
126 • SELECT IF (SELECT COUNT(*) FROM INFORMATION_SCHEMA.SCHEMATA WHERE SCHEMA_NAME = 'PetPals') > 0, 'PetPals database already exists.', 'PetPals database created successfully.' ) AS Message;
127
128 • SELECT Name, Age, Breed, Type
129   FROM Pets
130   WHERE AvailableForAdoption = 1;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

Name	Age	Breed	Type
Buddy	3	Labrador Retriever	Dog
Whiskers	2	Siamese	Cat
Snowball	4	Persian	Cat
Rocky	2	German Shepherd	Dog
Luna	1	Maine Coon	Cat
Milo	2	Ragdoll	Cat
Bailey	6	Beagle	Dog
Oliver	3	Russian Blue	Cat

Pets 46

34°C Sunny

Search

12:11 18-04-2024 ENG IN

6. Write an SQL query that retrieves the names of participants (shelters and adopters) registered for a specific adoption event. Use a parameter to specify the event ID. Ensure that the query joins the necessary tables to retrieve the participant names and types.

MySQL Workbench

case x

File Edit View Query Database Server Tools Scripting Help

Query 1 x

123
124 • CREATE DATABASE IF NOT EXISTS PetPals;
125
126 • SELECT IF((SELECT COUNT(*) FROM INFORMATION_SCHEMA.SCHEMATA WHERE SCHEMA_NAME = 'PetPals') > 0, 'PetPals database already exists.', 'PetPals database created successfully.') AS Message;
127
128 • SELECT Name, Age, Breed, Type
129 FROM Pets
130 WHERE AvailableForAdoption = 1;
131
132 • SELECT Participants.ParticipantName
133 FROM Participants
134 JOIN AdoptionEvents ON Participants.EventID = AdoptionEvents.EventID
135 WHERE AdoptionEvents.EventID = 3;
136
137
138 DELIMITER //
139
140 • CREATE PROCEDURE UpdateShelterInfo (
141 IN ShelterIDParam INT,
142 IN NewName VARCHAR(255),
143);

Result Grid | Export: | Wrap Cell Content: |

ParticipantName

Jay
Ananya

Result 45 x

34°C Sunny

Search

12:09 ENG IN 18-04-2024

7. Create a stored procedure in SQL that allows a shelter to update its information (name and location) in the "Shelters" table. Use parameters to pass the shelter ID and the new information. Ensure that the procedure performs the update and handles potential errors, such as an invalid shelter ID.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Query 1

135 WHERE AdoptionEvents.EventID = 3;

136

137

138 DELIMITER //

139

140 CREATE PROCEDURE UpdateShelterInfo (

141 IN ShelterIDParam INT,

142 IN NewName VARCHAR(255),

143 IN NewLocation VARCHAR(255)

144)

145 BEGIN

146 UPDATE Shelters

147 SET Name = NewName,

148 Location = NewLocation

149 WHERE ShelterID = ShelterIDParam;

150

151 END //

152

153

154

155 DELIMITER ;

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1100

1

8. Write an SQL query that calculates and retrieves the total donation amount for each shelter (by shelter name) from the "Donations" table. The result should include the shelter name and the total donation amount. Ensure that the query handles cases where a shelter has received no donations.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Query 1 ×

144)
145 BEGIN
146 UPDATE Shelters
147 SET Name = NewName,
148 Location = NewLocation
149 WHERE ShelterID = ShelterIDParam;
150 END //
151
152 DELIMITER ;
153
154 • SELECT s.Name AS ShelterName, IFNULL(SUM(d.DonationAmount), 0) AS TotalDonationAmount
155 FROM Shelters s
156 LEFT JOIN Donations d ON s.Name = d.DonorName
157 GROUP BY s.Name;|
158
159 • CREATE TABLE Owners (
160 OwnerID INT AUTO_INCREMENT PRIMARY KEY,
161 Name VARCHAR(255) NOT NULL,
162 Email VARCHAR(255)
163);

Result Grid | Filter Rows | Export: | Wrap Cell Content: |

ShelterName	TotalDonationAmount
Happy Shelter	0.00
Glory Shelter	0.00
Joy Shelter	0.00
Fluffy Shelter	0.00
Cute Shelter	0.00

Result 44 ×

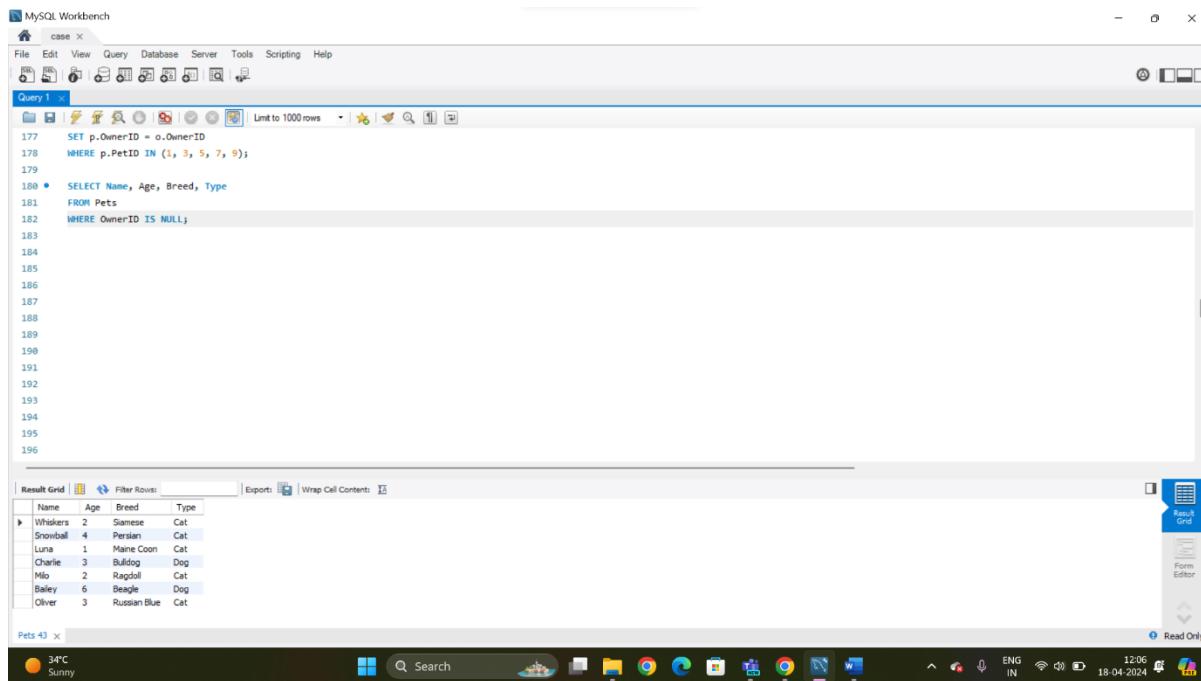
34°C Sunny

Search

1207 ENG IN

18-04-2024

9. Write an SQL query that retrieves the names of pets from the "Pets" table that do not have an owner (i.e., where "OwnerID" is null). Include the pet's name, age, breed, and type in the result set.



```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Query 1
177 SET p.OwnerID = o.OwnerID
178 WHERE p.PetID IN (1, 3, 5, 7, 9);
179
180 • SELECT Name, Age, Breed, Type
181 FROM Pets
182 WHERE OwnerID IS NULL;
183
184
185
186
187
188
189
190
191
192
193
194
195
196

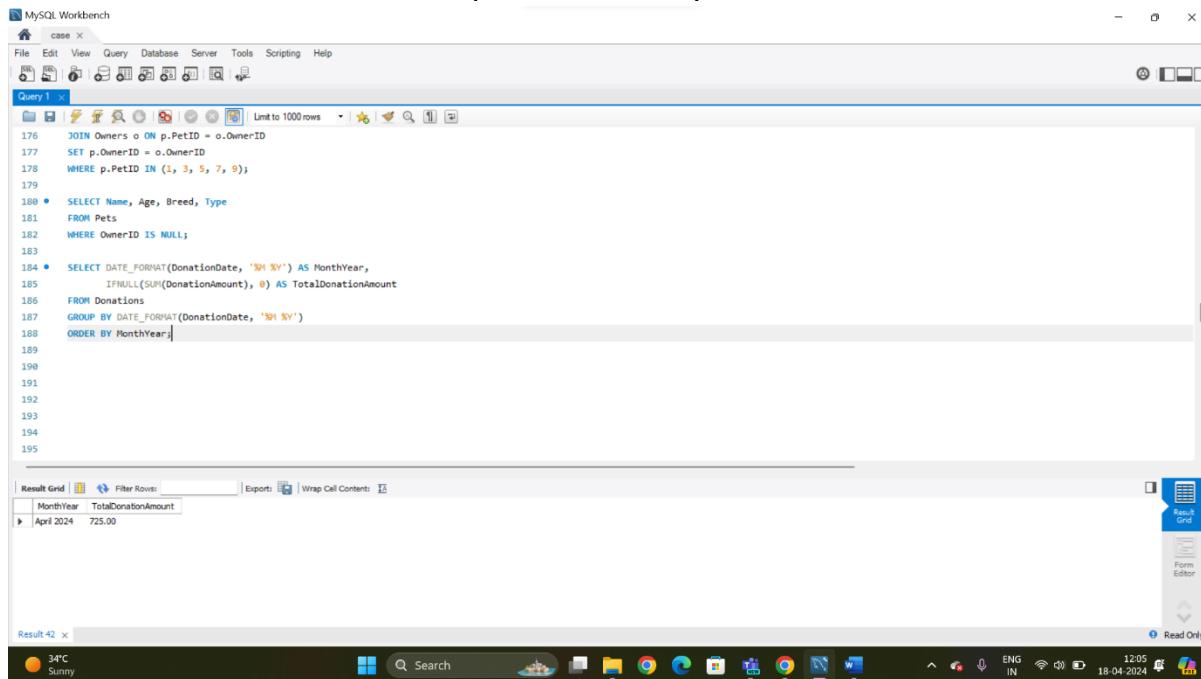
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid Form Editor Read Only

Name	Age	Breed	Type
Whiskers	2	Siamese	Cat
Snowball	4	Persian	Cat
Luna	1	Maine Coon	Cat
Charlie	3	Bulldog	Dog
Milo	2	Ragdoll	Cat
Bailey	6	Beagle	Dog
Oliver	3	Russian Blue	Cat

Pets 43 ×

10. Write an SQL query that retrieves the total donation amount for each month and year (e.g., January 2023) from the "Donations" table. The result should include the month-year and the corresponding total donation amount. Ensure that the query handles cases where no donations were made in a specific month-year.



```

MySQL Workbench
File Edit View Query Database Server Tools Scripting Help
Query 1
176 JOIN Owners o ON p.OwnerID = o.OwnerID
177 SET p.OwnerID = o.OwnerID
178 WHERE p.PetID IN (1, 3, 5, 7, 9);
179
180 • SELECT Name, Age, Breed, Type
181 FROM Pets
182 WHERE OwnerID IS NULL;
183
184 • SELECT DATE_FORMAT(DonationDate, '%M %Y') AS MonthYear,
185   IFNULL(SUM(DonationAmount), 0) AS TotalDonationAmount
186 FROM Donations
187 GROUP BY DATE_FORMAT(DonationDate, '%M %Y')
188 ORDER BY MonthYear;
189
190
191
192
193
194
195
196

```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid Form Editor Read Only

MonthYear	TotalDonationAmount
April 2024	725.00

Result 42 ×

11. Retrieve a list of distinct breeds for all pets that are either aged between 1 and 3 years or older than 5 years.

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query retrieves distinct breeds for pets aged between 1 and 3 years or older than 5 years. The result grid displays the following breeds:

Breed
Labrador Retriever
Siamese
German Shepherd
Maine Coon
Bulldog
Ragdoll
Beagle
Russian Blue

12. Retrieve a list of pets and their respective shelters where the pets are currently available for adoption.

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query retrieves a list of pets and their respective shelters, filtered for those available for adoption. The result grid displays the following data:

PetName	Breed	Type	ShelterName
Buddy	Labrador Retriever	Dog	Happy Shelter
Whiskers	Siamese	Cat	Happy Shelter
Snowball	Persian	Cat	Glory Shelter
Rocky	German Shepherd	Dog	Glory Shelter
Luna	Maine Coon	Cat	Cute Shelter
Milo	Ragdoll	Cat	Joy Shelter
Bailey	Beagle	Dog	Fluffy Shelter
Oliver	Russian Blue	Cat	Fluffy Shelter

13. Find the total number of participants in events organized by shelters located in specific city. Example: City=Chennai

The screenshot shows the MySQL Workbench interface. In the top window, titled 'Query 1', there is a SQL script. The script includes several JOIN clauses and a WHERE clause filtering by location ('ooty'). It also contains a COUNT(*) function to calculate the total number of participants. The result grid below shows one row with the value '2'. The bottom window, titled 'Result 38', displays the system tray with weather information (34°C, Sunny) and system status.

```
192 WHERE (Age BETWEEN 1 AND 5) OR (Age > 5);
193
194 • SELECT p.Name AS PetName, p.Breed, p.Type, s.Name AS ShelterName
195 FROM Pets p
196 JOIN Shelters s ON p.ShelterID = s.ShelterID
197 WHERE p.AvailableForAdoption = 1;
198
199 • SELECT COUNT(*) AS TotalParticipants
200 FROM Participants p
201 JOIN AdoptionEvents ae ON p.EventID = ae.EventID
202 JOIN Shelters s ON ae.Location = s.Location
203 WHERE s.Location = 'ooty';
204
205
206
207
208
209
210
211
```

TotalParticipants
2

Result 38 x

34°C Sunny

Search

1203 18-04-2024

14. Retrieve a list of unique breeds for pets with ages between 1 and 5 years.

The screenshot shows the MySQL Workbench interface. In the top window, titled 'Query 1', there is a SQL script. The script includes JOIN clauses and a WHERE clause filtering by age ('Age BETWEEN 1 AND 5'). It uses the DISTINCT keyword to retrieve unique breeds. The result grid below shows a list of breeds: Labrador Retriever, Siamese, Golden Retriever, Persian, German Shepherd, Maine Coon, Bulldog, and Ragdoll. The bottom window, titled 'Result 37', displays the system tray with weather information (34°C, Sunny) and system status.

```
199 • SELECT COUNT(*) AS TotalParticipants
200 FROM Participants p
201 JOIN AdoptionEvents ae ON p.EventID = ae.EventID
202 JOIN Shelters s ON ae.Location = s.Location
203 WHERE s.Location = 'ooty';
204
205 • SELECT DISTINCT Breed
206 FROM Pets
207 WHERE Age BETWEEN 1 AND 5;
208
209
210
211
212
213
214
215
216
217
218
```

Breed
Labrador Retriever
Siamese
Golden Retriever
Persian
German Shepherd
Maine Coon
Bulldog
Ragdoll

Result 37 x

34°C Sunny

Search

1203 18-04-2024

15. Find the pets that have not been adopted by selecting their information from the 'Pet' table.

```

199 • SELECT COUNT(*) AS TotalParticipants
200   FROM Participants p
201   JOIN AdoptionEvents ae ON p.EventID = ae.EventID
202   JOIN Shelters s ON ae.Location = s.Location
203   WHERE s.Location = 'ooty';
204
205 • SELECT DISTINCT Breed
206   FROM Pets
207   WHERE Age BETWEEN 1 AND 5;
208
209 • SELECT *
210   FROM Pets
211   WHERE OwnerID IS NULL;
212
213
214 • CREATE TABLE adoptedpets (
215     AdoptedPetID INT AUTO_INCREMENT PRIMARY KEY,
216     PetID INT,
217     OwnerID INT,
218     FOREIGN KEY (PetID) REFERENCES Pets(PetID),
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420

```

The screenshot shows the MySQL Workbench interface with a query editor window containing the provided SQL code. Below the editor is a result grid showing the output of the query. The result grid has columns: PetID, Name, Age, Breed, Type, AvailableForAdoption, OwnerID, and ShelterID. The data includes 10 rows of pet information. At the bottom of the screen, there is a taskbar with various icons and system status indicators.

PetID	Name	Age	Breed	Type	AvailableForAdoption	OwnerID	ShelterID
2	Whiskers	2	Siamese	Cat	1	NULL	1
4	Snowball	4	Persian	Cat	1	NULL	2
6	Luna	1	Maine Coon	Cat	1	NULL	5
7	Charlie	3	Bulldog	Dog	0	NULL	3
8	Milo	2	Ragdoll	Cat	1	NULL	3
9	Baley	6	Beagle	Dog	1	NULL	4
10	Oliver	3	Russian Blue	Cat	1	NULL	4
11	NULL	NULL	NULL	NULL	NULL	NULL	NULL
12	NULL	NULL	NULL	NULL	NULL	NULL	NULL
13	NULL	NULL	NULL	NULL	NULL	NULL	NULL

16. Retrieve the names of all adopted pets along with the adopter's name from the 'Adoption' and 'Owner' tables.

```

211 WHERE OwnerID IS NULL;
212
213
214 • CREATE TABLE adoptedpets (
215     AdoptedPetID INT AUTO_INCREMENT PRIMARY KEY,
216     PetID INT,
217     OwnerID INT,
218     FOREIGN KEY (PetID) REFERENCES Pets(PetID),
219     FOREIGN KEY (OwnerID) REFERENCES Owners(OwnerID)
220 );
221
222 • INSERT INTO adoptedpets (PetID, OwnerID)
223     SELECT PetID, OwnerID
224     FROM Pets
225     WHERE OwnerID IS NOT NULL;
226
227 • SELECT p.Name AS PetName, o.Name AS OwnerName
228     FROM adoptedpets ap
229     JOIN owners o ON ap.OwnerID = o.OwnerID
230     JOIN pets p ON ap.PetID = p.PetID;
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420

```

The screenshot shows the MySQL Workbench interface with a query editor window containing the provided SQL code. Below the editor is a result grid showing the output of the query. The result grid has columns: PetName and OwnerName. The data includes 3 rows of pet and owner names. At the bottom of the screen, there is a taskbar with various icons and system status indicators.

PetName	OwnerName
Buddy	Rahul Sharma
Max	Aryan Gupta
Rocky	Vikram Verma

17. Retrieve a list of all shelters along with the count of pets currently available for adoption in each shelter.

```

220
221
222 • INSERT INTO adoptedpets (PetID, OwnerID)
223     SELECT PetID, OwnerID
224     FROM Pets
225     WHERE OwnerID IS NOT NULL;
226
227 •     SELECT p.Name AS PetName, o.Name AS OwnerName
228     FROM adoptedpets ap
229     JOIN owners o ON ap.OwnerID = o.OwnerID
230     JOIN pets p ON ap.PetID = p.PetID;
231
232 •     SELECT s.Name AS ShelterName, COUNT(p.PetID) AS AvailablePetsCount
233     FROM Shelters s
234     LEFT JOIN Pets p ON s.ShelterID = p.ShelterID
235     WHERE p.AvailableForAdoption = 1 OR p.AvailableForAdoption IS NULL
236     GROUP BY s.ShelterID;
237
238 •     SELECT p1.Name AS Pet1Name, p1.Breed AS Breed, p2.Name AS Pet2Name, s.Name AS ShelterName
239     FROM Pets p1
240     LEFT JOIN Pets p2 ON p1.ShelterID = p2.ShelterID AND p1.Breed = p2.Breed AND p1.PetID < p2.PetID
241     JOIN Shelters s ON p1.ShelterID = s.ShelterID
242     ORDER BY p1.ShelterID, p1.Breed, p1.PetID;
243
244 •     SELECT s.Name AS ShelterName, ae.EventName AS EventName
245     FROM Shelters s
246     CROSS JOIN AdoptionEvents ae;
247
248 •     SELECT s.Name AS ShelterName
249     FROM Shelters s
250     LEFT JOIN Pets p ON s.ShelterID = p.ShelterID
251     LEFT JOIN AdoptedPets ap ON p.PetID = ap.PetID

```

Result Grid

ShelterName	AvailablePetsCount
Happy Shelter	2
Glory Shelter	2
Joy Shelter	1
Fluffy Shelter	2
Cute Shelter	1

18. Find pairs of pets from the same shelter that have the same breed.

```

232 •     SELECT s.Name AS ShelterName, COUNT(p.PetID) AS AvailablePetsCount
233     FROM Shelters s
234     LEFT JOIN Pets p ON s.ShelterID = p.ShelterID
235     WHERE p.AvailableForAdoption = 1 OR p.AvailableForAdoption IS NULL
236     GROUP BY s.ShelterID;
237
238 •     SELECT p1.Name AS Pet1Name, p1.Breed AS Breed, p2.Name AS Pet2Name, s.Name AS ShelterName
239     FROM Pets p1
240     LEFT JOIN Pets p2 ON p1.ShelterID = p2.ShelterID AND p1.Breed = p2.Breed AND p1.PetID < p2.PetID
241     JOIN Shelters s ON p1.ShelterID = s.ShelterID
242     ORDER BY p1.ShelterID, p1.Breed, p1.PetID;
243
244 •     SELECT s.Name AS ShelterName, ae.EventName AS EventName
245     FROM Shelters s
246     CROSS JOIN AdoptionEvents ae;
247
248 •     SELECT s.Name AS ShelterName
249     FROM Shelters s
250     LEFT JOIN Pets p ON s.ShelterID = p.ShelterID
251     LEFT JOIN AdoptedPets ap ON p.PetID = ap.PetID

```

Result Grid

Pet1Name	Breed	Pet2Name	ShelterName
Buddy	Labrador Retriever	Mall	Happy Shelter
Whiskers	Siamese	Mall	Happy Shelter
Rocky	German Shepherd	Mall	Glory Shelter
Snowball	Persian	Mall	Glory Shelter
Charlie	Bulldog	Mall	Joy Shelter
Milo	Ragdoll	Mall	Joy Shelter
Bailey	Beagle	Mall	Fluffy Shelter
Oliver	Russian Blue	Mall	Fluffy Shelter

19. List all possible combinations of shelters and adoption events.

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query retrieves shelter names and adoption event names from various tables. The result grid displays the data in two columns: ShelterName and EventName.

```
232 • SELECT s.Name AS ShelterName, COUNT(p.PetID) AS AvailablePetsCount
233   FROM Shelters s
234   LEFT JOIN Pets p ON s.ShelterID = p.ShelterID
235   WHERE p.AvailableForAdoption = 1 OR p.AvailableForAdoption IS NULL
236   GROUP BY s.ShelterID;
237
238 • SELECT p1.Name AS Pet1Name, p1.Breed AS Breed, p2.Name AS Pet2Name, s.Name AS ShelterName
239   FROM Pets p1
240   LEFT JOIN Pets p2 ON p1.ShelterID = p2.ShelterID AND p1.Breed = p2.Breed AND p1.PetID < p2.PetID
241   JOIN Shelters s ON p1.ShelterID = s.ShelterID
242   ORDER BY p1.ShelterID, p1.Breed, p1.PetID;
243
244 • SELECT s.Name AS ShelterName, ae.EventName AS EventName
245   FROM Shelters s
246   CROSS JOIN AdoptionEvents ae;
247
248 • SELECT s.Name AS ShelterName
249   FROM Shelters s
250   LEFT JOIN Pets p ON s.ShelterID = p.ShelterID
251   LEFT JOIN AdoptedPets ap ON p.PetID = ap.PetID
```

ShelterName	EventName
Joy Shelter	Cat Adoption Fair
Glory Shelter	Cat Adoption Fair
Fluffy Shelter	Dog Meet-and-Greet
Cute Shelter	Dog Meet-and-G...
Fluffy Shelter	Dog Meet-and-G..., Dog Meet-and-Greet
Joy Shelter	Dog Meet-and-G...
Glory Shelter	Dog Meet-and-G...
Happy Shelter	Dog Meet-and-G...
Fluffy Shelter	Dog Meet-and-G...

20. Determine the shelter that has the highest number of adopted pets.

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query retrieves shelter names and the count of adopted pets, ordered by the count in descending order with a limit of 1. The result grid shows one row for the Happy Shelter.

```
247
248 • SELECT s.Name AS ShelterName
249   FROM Shelters s
250   LEFT JOIN Pets p ON s.ShelterID = p.ShelterID
251   LEFT JOIN AdoptedPets ap ON p.PetID = ap.PetID
252   GROUP BY s.ShelterID
253   ORDER BY COUNT(ap.PetID) DESC
254   LIMIT 1;
```

ShelterName
Happy Shelter