

Name: Diya Jain

Div: TY-3

Roll No: 26

Batch: B

Subject: DWM

Experiment No 02

AIM : Implementation of OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot for the above problem statement (experiment 1).

Problem statement :

Design a data warehouse for a regional weather bureau. The weather bureau has about 100 probs, which are scattered throughout various land and ocean locations in the region to collect basic weather data, including air pressure, temperature and precipitation at each hour. All data are sent to the central station, which has collected such data for more than 10 years. Design Star schema and Snowflake schema such that it should facilitate efficient querying and online analytical processing and derive general weather patterns in multidimensional space. Explain all aspects of the diagram. Design Star and Snowflake schema for above case.

THEORY :

Online Analytical Processing (OLAP) is a technology that enables analysts to extract and view data from different points of view. OLAP operations are designed to facilitate multidimensional analysis of business data for decision-making purposes. The core OLAP operations include:

1. Slice Operation

Slicing involves selecting a specific value for one dimension and viewing the resulting "slice" of the data cube. It's like cutting a slice from a cube, reducing the dimensionality by one while keeping all other dimensions intact.

2. Dice Operation

Dicing involves selecting a subcube by applying restrictions on two or more dimensions simultaneously. Unlike slicing which reduces dimensionality, dicing maintains the same number of dimensions but with fewer values in each.

3. Roll-up (Aggregation)

Roll-up performs aggregation on a data cube by climbing up a concept hierarchy for a dimension or by reducing dimensions. It allows for summarizing data at different levels of granularity, moving from detailed to more summarized views.

4. Drill-down

Drill-down is the opposite of roll-up. It navigates from less detailed data to more detailed data, descending the concept hierarchy for a dimension or introducing additional dimensions for more granular analysis.

5. Pivot (Rotate)

Pivoting rotates the data axes to provide an alternative presentation of data. It helps in viewing the same information from different perspectives by reorganizing the multidimensional view.

Code :

-- Slice: Get all temperature measurements for a specific location (LocationID = 1)

SELECT

t.Hour,

l.LocationName,

mt.TypeName,

wm.Value

FROM

WeatherMeasurements wm

JOIN

TimeDim t ON wm.TimeID = t.TimeID

JOIN

LocationDim l ON wm.LocationID = l.LocationID

JOIN

MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID

WHERE

l.LocationID = 1 AND mt.TypeName = 'Temperature';

-- Dice: Get all measurements for a specific location, time range, and measurement types

SELECT

t.Hour,

l.LocationName,

p.ProbeModel,

mt.TypeName,

mt.Unit,

```

    wm.Value

FROM

    WeatherMeasurements wm

JOIN

    TimeDim t ON wm.TimeID = t.TimeID

JOIN

    LocationDim l ON wm.LocationID = l.LocationID

JOIN

    ProbeDim p ON wm.ProbeID = p.ProbeID

JOIN

    MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID

WHERE

    l.LocationID = 1 AND

    t.Hour BETWEEN 9 AND 11 AND

    mt.TypeName IN ('Temperature', 'Pressure');

-- Roll-up: Aggregate temperature data by location type

SELECT

    l.LocationType,

    mt.TypeName,

    AVG(wm.Value) as AvgValue,

    MIN(wm.Value) as MinValue,

    MAX(wm.Value) as MaxValue,

    COUNT(*) as NumberOfReadings

FROM

```

```
WeatherMeasurements wm
JOIN
LocationDim l ON wm.LocationID = l.LocationID
JOIN
MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
WHERE
mt.TypeName = 'Temperature'
GROUP BY
l.LocationType, mt.TypeName;
```

-- Drill-down: From region level to specific locations

```
SELECT
r.RegionName,
sr.SubRegionName,
l.LocationName,
mt.TypeName,
AVG(wm.Value) as AvgValue
FROM
WeatherMeasurements wm
JOIN
LocationDim l ON wm.LocationID = l.LocationID
JOIN
SubRegionDim sr ON l.SubRegionID = sr.SubRegionID
JOIN
RegionDim r ON sr.RegionID = r.RegionID
```

JOIN

MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID

WHERE

mt.TypeName = 'Temperature'

GROUP BY

r.RegionName, sr.SubRegionName, l.LocationName, mt.TypeName

ORDER BY

r.RegionName, sr.SubRegionName, l.LocationName;

-- Pivot: Transform rows of measurement types into columns

SELECT

t.Hour,

l.LocationName,

MAX(CASE WHEN mt.TypeName = 'Temperature' THEN wm.Value END) as Temperature,

MAX(CASE WHEN mt.TypeName = 'Pressure' THEN wm.Value END) as Pressure,

MAX(CASE WHEN mt.TypeName = 'Rainfall' THEN wm.Value END) as Rainfall

FROM

WeatherMeasurements wm

JOIN

TimeDim t ON wm.TimeID = t.TimeID

JOIN

LocationDim l ON wm.LocationID = l.LocationID

JOIN

MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID

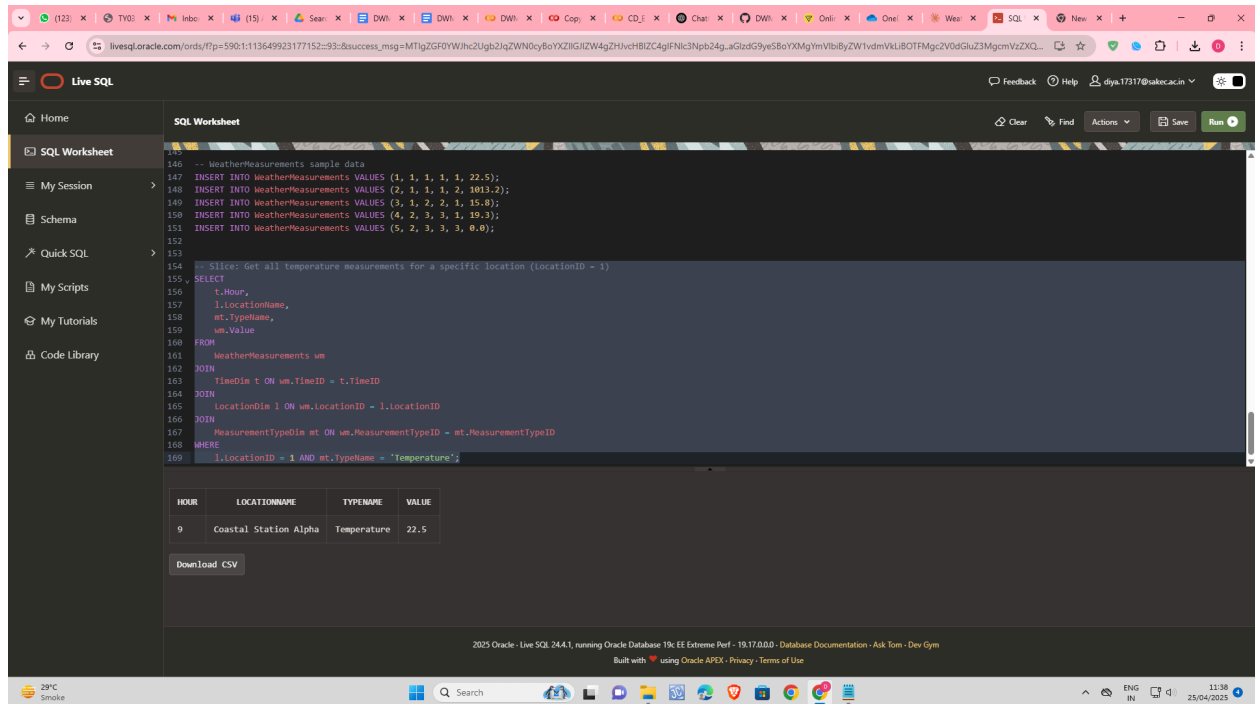
GROUP BY

t.Hour, l.LocationName

ORDER BY

t.Hour, l.LocationName;

PROGRAM OUTPUT :



The screenshot displays the Live SQL web application interface. The left sidebar contains navigation links: Home, SQL Worksheet, My Session, Schema, Quick SQL, My Scripts, My Tutorials, and Code Library. The main area shows a SQL Worksheet with a query that inserts sample data into a table named 'WeatherMeasurements' and then queries it. The query filters for 'Temperature' measurements at 'Coastal Station Alpha'.

```
146 -- WeatherMeasurements sample data
147 INSERT INTO WeatherMeasurements VALUES (1, 1, 1, 1, 1, 22.5);
148 INSERT INTO WeatherMeasurements VALUES (2, 1, 1, 1, 2, 1013.2);
149 INSERT INTO WeatherMeasurements VALUES (3, 1, 2, 2, 1, 15.8);
150 INSERT INTO WeatherMeasurements VALUES (4, 2, 3, 3, 1, 19.3);
151 INSERT INTO WeatherMeasurements VALUES (5, 2, 3, 3, 3, 0.8);
152
153
154 -- Slice: Get all temperature measurements for a specific location (locationID = 1)
155 SELECT
156     t.Hour,
157     l.LocationName,
158     mt.TypeName,
159     mv.Value
160 FROM
161     WeatherMeasurements mv
162 JOIN
163     TimeDim t ON mv.TimeID = t.TimeID
164 JOIN
165     LocationDim l ON mv.LocationID = l.LocationID
166 JOIN
167     MeasurementTypeDim mt ON mv.MeasurementTypeID = mt.MeasurementTypeID
168 WHERE
169     l.LocationID = 1 AND mt.TypeName = 'Temperature';
```

The output of the query is displayed in a table with the following data:

HOUR	LOCATIONNAME	TYPENAME	VALUE
9	Coastal Station Alpha	Temperature	22.5

Below the table is a 'Download CSV' button. At the bottom of the interface, a footer note states: '2025 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with using Oracle APEX - Privacy - Terms of Use'.

Live SQL

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

185

-- Dice: Get all measurements for a specific location, time range, and measurement types

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

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

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

HOUR	LOCATIONNAME	PROBEMODEL	TYPENAME	UNIT	VALUE
9	Coastal Station Alpha	WeatherMaster X1	Temperature	°C	22.5
9	Coastal Station Alpha	WeatherMaster X1	Pressure	hPa	1013.2

Download CSV

2 rows selected.

2025 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym

Built with using Oracle APEX - Privacy - Terms of Use

28°C

Smoke

Live SQL

Home

SQL Worksheet

My Session

Schema

Quick SQL

My Scripts

My Tutorials

Code Library

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

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

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

Live SQL

SQL Worksheet

```
-- Drill-down: From region level to specific locations
215 SELECT
216   r.RegionName,
217   sr.SubRegionName,
218   l.LocationName,
219   mt.TypeName,
220   AVG(wm.Value) as AvgValue
221 FROM
222   WeatherMeasurements wm
223 JOIN
224   locationDim l ON wm.locationID = l.locationID
225 JOIN
226   SubRegionDim sr ON l.SubRegionID = sr.SubRegionID
227 JOIN
228   RegionDim r ON sr.RegionID = r.RegionID
229 JOIN
230   MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
231 WHERE
232   mt.TypeName = 'Temperature'
233 GROUP BY
234   r.RegionName, sr.SubRegionName, l.LocationName, mt.TypeName
235 ORDER BY
236   r.RegionName, sr.SubRegionName, l.LocationName;
```

REGIONNAME	SUBREGIONNAME	LOCATIONNAME	TYPENAME	AVGVALUE
Mountain	Alpine Zone	Mountain Peak Beta	Temperature	15.8
Ocean	Deep Sea	Ocean Buoy Charlie	Temperature	19.3
West Coast	Coastal Area	Coastal Station Alpha	Temperature	22.5

Download CSV

3 rows selected.

2025 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with using Oracle APEX - Privacy - Terms of Use

Live SQL

SQL Worksheet

```
-- Pivot: Transform rows of measurement types into columns
238 SELECT
239   t.Hour,
240   l.LocationName,
241   MAX(CASE WHEN mt.TypeName = 'Temperature' THEN wm.Value END) as Temperature,
242   MAX(CASE WHEN mt.TypeName = 'Pressure' THEN wm.Value END) as Pressure,
243   MAX(CASE WHEN mt.TypeName = 'Rainfall' THEN wm.Value END) as Rainfall
244 FROM
245   WeatherMeasurements wm
246 JOIN
247   TimeDim t ON wm.TimeID = t.TimeID
248 JOIN
249   locationDim l ON wm.locationID = l.locationID
250 JOIN
251   MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
252 GROUP BY
253   t.Hour, l.LocationName
254 ORDER BY
255   t.Hour, l.LocationName;
```

HOUR	LOCATIONNAME	TEMPERATURE	PRESSURE	RAINFALL
9	Coastal Station Alpha	22.5	1013.2	-
9	Mountain Peak Beta	15.8	-	-
10	Ocean Buoy Charlie	19.3	-	0

Download CSV

3 rows selected.

2025 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with using Oracle APEX - Privacy - Terms of Use

Review Question :

1. What is the difference between Slice and Dice operations in OLAP?

Ans : Slice selects one specific value for one dimension, reducing dimensionality by one (like cutting a single slice from a cube). Dice selects multiple values across multiple dimensions, creating a smaller subcube while maintaining all dimensions.

2. How does Roll-up operation help in summarizing large volumes of data?

Ans : Roll-up summarizes large volumes of data by aggregating along concept hierarchies (e.g., daily→monthly→quarterly), reducing dimensionality, improving query performance, revealing high-level patterns, and supporting strategic decision-making by eliminating excessive detail.

3. Example of Pivoting providing clearer insight than traditional tabular view:

Ans : In retail sales analysis, pivoting transforms data from a linear list (month, region, product, sales) into a cross-tabulation with regions as rows and months as columns for each product category. This format instantly reveals regional performance patterns, seasonal trends, and product category comparisons that would be difficult to identify in traditional tabular format.

GITHUB LINK : <https://github.com/DiyaJain09/DWM-EXPERIMENTS.git>