

REPORTE DE PRÁCTICA NO. 3.3

Estrategia Bottom-Up

ALUMNO: Armando Oswaldo Perez Romero
Dr. Eduardo Cornejo-Velázquez



Introducción

En el desarrollo de bases de datos, la estrategia Bottom-Up es fundamental para diseñar esquemas robustos y escalables, partiendo desde las entidades más específicas hasta las relaciones más complejas. Durante esta práctica, se implementó un sistema de gestión académica e investigativa utilizando este enfoque.

En esta práctica se llevaron a cabo procesos de extracción, transformación y carga (ETL) para integrar tablas de tres bases de datos distintas. Estas bases de datos, que pueden diferir en estructura, formato y ubicación, representan un desafío en términos de interoperabilidad y consistencia de datos. A través de técnicas como la fragmentación vertical, así como el uso de herramientas y lenguajes de consulta especializados, se logró consolidar la información en un único entorno para su posterior análisis y explotación.

2. Marco teórico

SQL

El lenguaje de consulta estructurado (SQL, por sus siglas en inglés) es un lenguaje estándar empleado para administrar y manejar bases de datos relacionales. Mediante SQL, los usuarios tienen la posibilidad de realizar diversas operaciones en las bases de datos, tales como añadir, actualizar, eliminar y consultar información. Este lenguaje es fundamental para interactuar con sistemas de gestión de bases de datos (SGBD) relacionales, como MySQL, PostgreSQL, SQL Server y Oracle, entre otros.

SQL es un lenguaje declarativo, lo que significa que el usuario especifica qué datos desea obtener o modificar, y el SGBD se encarga de determinar la mejor manera de ejecutar la consulta.

Referencia: Escofet, C. M. (2002). El lenguaje SQL. UOC, la universidad virtual.

MySQL

MySQL es un sistema de gestión de bases de datos (SGBD) de código abierto ampliamente utilizado en el desarrollo de aplicaciones web, sistemas empresariales y diversas plataformas que requieren almacenamiento y manejo eficiente de datos. Se destaca por su rapidez, fiabilidad y facilidad de uso, además de ser compatible con múltiples sistemas operativos, como Windows, Linux y macOS.

Algunas características clave de MySQL incluyen:

MySQL es utilizado por empresas y organizaciones de todo el mundo debido a su flexibilidad y robustez. Además, cuenta con una comunidad activa que contribuye a su desarrollo y mejora continua.

Referencia: Santillán, L. A. C., Ginestà, M. G., Mora, Ó. P. (2014). Bases de datos en MySQL. Universitat oberta de Catalunya.

Procesos ETL (Extract, Transform, Load)

Los procesos ETL (Extracción, Transformación y Carga) son fundamentales en el ámbito de la integración de datos y el almacenamiento en data warehouses. Estos procesos permiten mover datos desde múltiples fuentes heterogéneas a un repositorio centralizado, donde pueden ser utilizados para análisis y reporting.

- **Extracción (Extract):** En esta fase, los datos se recopilan de diversas fuentes, como bases de datos relacionales, archivos planos, APIs, o sistemas legacy. La extracción puede ser completa (todos los datos) o incremental (solo los cambios desde la última extracción).
- **Transformación (Transform):** Los datos extraídos se someten a una serie de operaciones para limpiarlos, normalizarlos y estructurarlos según los requisitos del destino. Esto incluye la eliminación de duplicados, la conversión de formatos, la aplicación de reglas de negocio y la agregación de datos.
- **Carga (Load):** Finalmente, los datos transformados se cargan en el sistema de destino, que puede ser un data warehouse, un data lake o una base de datos analítica. La carga puede ser en modo *full refresh* (sobrescribir todos los datos) o *incremental* (agregar solo los nuevos datos).

Los procesos ETL son esenciales para garantizar la calidad, consistencia y disponibilidad de los datos en entornos empresariales. **Referencia:** Kimball, R., Ross, M. (2013). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. Wiley.

SELECT + INTO FILE en Bases de Datos

La sentencia `SELECT + INTO FILE` es una operación común en bases de datos que permite exportar el resultado de una consulta a un archivo externo. Esta funcionalidad es útil para generar informes, compartir datos con otros sistemas o realizar copias de seguridad parciales.

- **Sintaxis básica:** La estructura típica de esta operación es:

```
SELECT columnas
INTO OUTFILE 'ruta_del_archivo'
FROM tabla
WHERE condiciones;
```

- **Formatos soportados:** Dependiendo del sistema de gestión de bases de datos (DBMS), el archivo generado puede estar en formatos como CSV, TXT o incluso binario.
- **Aplicaciones comunes:** Esta operación se utiliza en escenarios como la exportación de datos para análisis externos, la migración de datos entre sistemas o la creación de archivos de configuración.

Es importante tener en cuenta los permisos de escritura en el servidor y las restricciones de seguridad al utilizar esta operación. **Referencia:** Elmasri, R., Navathe, S. B. (2016). Fundamentals of Database Systems. Pearson.

LOAD en Bases de Datos

La operación LOAD se utiliza para cargar datos desde un archivo externo a una tabla en una base de datos. Esta operación es común en procesos de integración de datos, migraciones y actualizaciones masivas.

- **Sintaxis básica:** La estructura típica de esta operación es:

```
LOAD DATA INFILE 'ruta_del_archivo'
INTO TABLE nombre_tabla
FIELDS TERMINATED BY 'delimitador'
LINES TERMINATED BY 'delimitador_de_linea';
```

- **Formatos soportados:** Los archivos de entrada suelen estar en formato CSV o TXT, aunque algunos DBMS admiten otros formatos.
- **Aplicaciones comunes:** Esta operación es útil en escenarios como la carga inicial de datos en un sistema, la actualización masiva de registros o la integración de datos desde fuentes externas.

Es fundamental asegurarse de que el formato del archivo coincida con la estructura de la tabla y de que se manejen adecuadamente los errores durante la carga. **Referencia:** Date, C. J. (2004). An Introduction to Database Systems. Addison-Wesley.

Estrategia Bottom-Up

La estrategia Bottom-Up (de abajo hacia arriba) es un enfoque de diseño y construcción de sistemas de almacenamiento de datos (data warehouses) propuesto principalmente por Ralph Kimball. Bajo este enfoque, el desarrollo se centra primero en la creación de data marts específicos para áreas particulares del negocio, y posteriormente estos se integran para formar un data warehouse corporativo más amplio.

- **Construcción de Data Marts:** Inicialmente, se diseñan y construyen pequeños data marts enfocados en procesos de negocio concretos, como ventas, inventarios o atención al cliente. Cada data mart es una colección de datos organizados de manera que sirvan de forma óptima a las necesidades analíticas de un área específica.
- **Integración de Data Marts:** A medida que se crean más data marts, se van integrando mediante la estandarización de dimensiones y medidas comunes. Esta integración permite construir una arquitectura de datos cohesiva a nivel organizacional.
- **Enfoque Incremental:** El enfoque Bottom-Up permite obtener beneficios analíticos de manera temprana, ya que los data marts individuales pueden ser implementados y utilizados rápidamente, mientras el data warehouse completo se desarrolla de manera progresiva.

La estrategia Bottom-Up es apreciada por su rapidez en la entrega de valor al negocio, su enfoque modular y su flexibilidad para adaptarse a cambios en los requisitos. Sin embargo, requiere una cuidadosa planificación para asegurar la consistencia de los datos entre los distintos data marts.

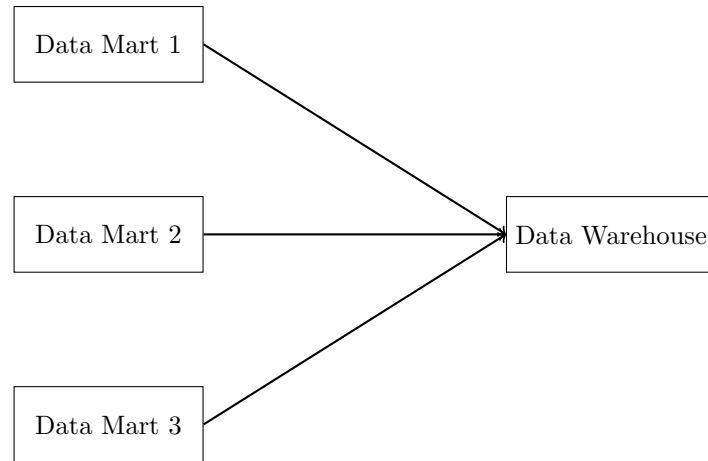


Figure 1: Integración de Data Marts en un Data Warehouse (Enfoque Bottom-Up)

Referencia: Kimball, R., Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley.

3. Herramientas empleadas

MySQL Workbench

MySQL Workbench es una herramienta visual de diseño y administración de bases de datos que facilita la creación, modificación y gestión de bases de datos MySQL. Ofrece un entorno integrado para diseñar esquemas de bases de datos, escribir consultas SQL, administrar usuarios y realizar tareas de mantenimiento. En esta práctica, se utilizó MySQL Workbench para implementar y probar la fragmentación vertical y horizontal, así como para ejecutar consultas y visualizar los resultados de manera eficiente.

Overleaf (Editor LaTeX Online)

Overleaf es un editor en línea basado en LaTeX que permite crear, editar y compartir documentos académicos, técnicos y científicos de alta calidad. Su interfaz intuitiva y su capacidad para compilar documentos en tiempo real lo convierten en una herramienta ideal para la redacción de informes, artículos y presentaciones. En este reporte, se utilizó Overleaf para la elaboración del documento, aprovechando sus plantillas y funcionalidades para dar formato al contenido de manera profesional.

ChatGPT

ChatGPT es un modelo de lenguaje desarrollado por OpenAI, capaz de comprender instrucciones en lenguaje natural y generar respuestas coherentes, explicaciones, fragmentos de código y soluciones a problemas técnicos. En esta práctica, ChatGPT se utilizó como apoyo para generar registros SQL, diseñar estructuras de tablas y normalizar bases de datos. Además, sirvió como herramienta de consulta para resolver dudas conceptuales y mejorar la eficiencia del desarrollo.

MySQL

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto ampliamente utilizado en aplicaciones empresariales y académicas. Permite almacenar, modificar y consultar grandes volúmenes de datos de manera estructurada mediante el uso del lenguaje SQL (Structured Query Language). En esta práctica, MySQL fue empleado para crear bases de datos, definir esquemas de tablas, insertar registros, realizar operaciones de fragmentación de datos (vertical y horizontal) y ejecutar consultas de prueba para validar el correcto funcionamiento de la estructura propuesta.

4. Desarrollo

Scripts de creación de nodos

```
1 CREATE DATABASE Local_Integrante;
2 USE Local_Integrante;
3 CREATE TABLE Persona
4 (
5     idPersona INT NOT NULL,
6     paterno CHAR(80) NOT NULL,
7     materno CHAR(80) NOT NULL,
8     nombre CHAR(120) NOT NULL,
9     PRIMARY KEY (idPersona)
10 );
11
12 CREATE TABLE Integrante
13 (
14     idIntegrante INT NOT NULL,
15     idPersona INT NOT NULL,
16     PRIMARY KEY (idIntegrante),
17     FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
18 );
19
20 CREATE TABLE CuerpoAcademico
21 (
22     idCuerpo INT NOT NULL,
23     nombre VARCHAR(250) NOT NULL,
24     idIntegrante INT NOT NULL,
25     PRIMARY KEY (idCuerpo),
26     FOREIGN KEY (idIntegrante) REFERENCES Integrante(idIntegrante)
27 );
28
29 CREATE TABLE Linea
30 (
31     idLinea INT NOT NULL,
32     nombre VARCHAR(120) NOT NULL,
33     descripcion VARCHAR(500) NOT NULL,
34     idCuerpo INT NOT NULL,
35     PRIMARY KEY (idLinea),
36     FOREIGN KEY (idCuerpo) REFERENCES CuerpoAcademico(idCuerpo)
37 );
38
39 CREATE TABLE IntegranteLinea
40 (
41     idIntegranteLinea INT NOT NULL,
42     vigente BOOLEAN NOT NULL,
43     inicio DATE NOT NULL,
44     termino DATE NOT NULL,
45     idLinea INT NOT NULL,
46     idIntegrante INT NOT NULL,
47     PRIMARY KEY (idIntegranteLinea),
48     FOREIGN KEY (idLinea) REFERENCES Linea(idLinea),
49     FOREIGN KEY (idIntegrante) REFERENCES Integrante(idIntegrante)
50 );
51
52 CREATE DATABASE Local_Investigador;
53 USE Local_Investigador;
54 CREATE TABLE Persona
55 (
56     idPersona INT NOT NULL,
57     paterno CHAR(80) NOT NULL,
58     materno CHAR(80) NOT NULL,
59     nombre CHAR(120) NOT NULL,
60     PRIMARY KEY (idPersona)
61 );
62
63 CREATE TABLE Investigador
```

```

64 (
65     idInvestigador INT NOT NULL,
66     orcid VARCHAR(30) NOT NULL,
67     email VARCHAR(150) NOT NULL,
68     movil VARCHAR(15) NOT NULL,
69     idPersona INT NOT NULL,
70     PRIMARY KEY (idInvestigador),
71     FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
72 );
73
74 CREATE TABLE Proyecto
75 (
76     idProyecto INT NOT NULL,
77     nombre VARCHAR(250) NOT NULL,
78     inicio DATE NOT NULL,
79     final DATE NOT NULL,
80     idInvestigador INT NOT NULL,
81     PRIMARY KEY (idProyecto),
82     FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
83 );
84
85 CREATE TABLE Produccion
86 (
87     idProduccion INT NOT NULL,
88     tipo VARCHAR(60) NOT NULL,
89     titulo VARCHAR(250) NOT NULL,
90     anio INT NOT NULL,
91     idInvestigador INT NOT NULL,
92     PRIMARY KEY (idProduccion),
93     FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
94 );
95
96 CREATE TABLE Adscripcion
97 (
98     idAdscripcion INT NOT NULL,
99     instituto VARCHAR(80) NOT NULL,
100     area VARCHAR(150) NOT NULL,
101     nombramiento VARCHAR(10) NOT NULL,
102     fechaingreso DATETIME NOT NULL,
103     idInvestigador INT NOT NULL,
104     PRIMARY KEY (idAdscripcion),
105     FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
106 );
107
108 CREATE TABLE Formacion
109 (
110     idFormacion INT NOT NULL,
111     grado VARCHAR(18) NOT NULL,
112     institucion VARCHAR(70) NOT NULL,
113     nombre VARCHAR(120) NOT NULL,
114     fechatermino DATE NOT NULL,
115     idInvestigador INT NOT NULL,
116     PRIMARY KEY (idFormacion),
117     FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
118 );
119
120 CREATE DATABASE Local_Profesor;
121 USE Local_Profesor;
122 CREATE TABLE Programa
123 (
124     idPrograma INT NOT NULL,
125     nivel VARCHAR(25) NOT NULL,
126     nombre VARCHAR(150) NOT NULL,
127     fechainicio DATE NOT NULL,
128     PRIMARY KEY (idPrograma)
129 );
130
131 CREATE TABLE Alumno

```



```

132 (
133     idAlumno INT NOT NULL,
134     nocuenta VARCHAR(25) NOT NULL,
135     paterno VARCHAR(80) NOT NULL,
136     materno VARCHAR(80) NOT NULL,
137     nombre VARCHAR(80) NOT NULL,
138     email VARCHAR(120) NOT NULL,
139     PRIMARY KEY (idAlumno)
140 );
141
142 CREATE TABLE Persona
143 (
144     idPersona INT NOT NULL,
145     paterno CHAR(80) NOT NULL,
146     materno CHAR(80) NOT NULL,
147     nombre CHAR(120) NOT NULL,
148     PRIMARY KEY (idPersona)
149 );
150
151 CREATE TABLE Profesor
152 (
153     idProfesor INT NOT NULL,
154     email VARCHAR(120) NOT NULL,
155     fechaingreso DATE NOT NULL,
156     idPersona INT NOT NULL,
157     PRIMARY KEY (idProfesor),
158     FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
159 );
160
161 CREATE TABLE Asignatura
162 (
163     idAsignatura INT NOT NULL,
164     semestre INT NOT NULL,
165     nombre VARCHAR(120) NOT NULL,
166     idPrograma INT NOT NULL,
167     PRIMARY KEY (idAsignatura),
168     FOREIGN KEY (idPrograma) REFERENCES Programa(idPrograma)
169 );
170
171 CREATE TABLE Curso
172 (
173     idCurso INT NOT NULL,
174     periodo VARCHAR(15) NOT NULL,
175     anio INT NOT NULL,
176     grupo INT NOT NULL,
177     idProfesor INT NOT NULL,
178     idAsignatura INT NOT NULL,
179     PRIMARY KEY (idCurso),
180     FOREIGN KEY (idProfesor) REFERENCES Profesor(idProfesor),
181     FOREIGN KEY (idAsignatura) REFERENCES Asignatura(idAsignatura)
182 );
183
184 CREATE TABLE AlumnoCurso
185 (
186     idAlumnoCurso INT NOT NULL,
187     calificacion FLOAT NOT NULL,
188     idCurso INT NOT NULL,
189     idAlumno INT NOT NULL,
190     PRIMARY KEY (idAlumnoCurso),
191     FOREIGN KEY (idCurso) REFERENCES Curso(idCurso),
192     FOREIGN KEY (idAlumno) REFERENCES Alumno(idAlumno)
193 );
194
195 -- Global --
196
197 CREATE DATABASE Global_u;
198 USE Global_u;
199 CREATE TABLE Programa

```

```

200 (
201     idPrograma INT NOT NULL,
202     nivel VARCHAR(25) NOT NULL,
203     nombre VARCHAR(150) NOT NULL,
204     fechainicio DATE NOT NULL,
205     PRIMARY KEY (idPrograma)
206 );
207
208 CREATE TABLE Alumno
209 (
210     idAlumno INT NOT NULL,
211     nocuenta VARCHAR(25) NOT NULL,
212     paterno VARCHAR(80) NOT NULL,
213     materno VARCHAR(80) NOT NULL,
214     nombre VARCHAR(80) NOT NULL,
215     email VARCHAR(120) NOT NULL,
216     PRIMARY KEY (idAlumno)
217 );
218
219 CREATE TABLE Persona
220 (
221     idPersona INT NOT NULL,
222     paterno CHAR(80) NOT NULL,
223     materno CHAR(80) NOT NULL,
224     nombre CHAR(120) NOT NULL,
225     PRIMARY KEY (idPersona)
226 );
227
228 CREATE TABLE Integrante
229 (
230     idIntegrante INT NOT NULL,
231     idPersona INT NOT NULL,
232     PRIMARY KEY (idIntegrante),
233     FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
234 );
235
236 CREATE TABLE CuerpoAcademico
237 (
238     idCuerpo INT NOT NULL,
239     nombre VARCHAR(250) NOT NULL,
240     idIntegrante INT NOT NULL,
241     PRIMARY KEY (idCuerpo),
242     FOREIGN KEY (idIntegrante) REFERENCES Integrante(idIntegrante)
243 );
244
245 CREATE TABLE Linea
246 (
247     idLinea INT NOT NULL,
248     nombre VARCHAR(120) NOT NULL,
249     descripcion VARCHAR(500) NOT NULL,
250     idCuerpo INT NOT NULL,
251     PRIMARY KEY (idLinea),
252     FOREIGN KEY (idCuerpo) REFERENCES CuerpoAcademico(idCuerpo)
253 );
254
255 CREATE TABLE Profesor
256 (
257     idProfesor INT NOT NULL,
258     email VARCHAR(120) NOT NULL,
259     fechaingreso DATE NOT NULL,
260     idPersona INT NOT NULL,
261     PRIMARY KEY (idProfesor),
262     FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
263 );
264
265 CREATE TABLE Investigador
266 (
267     idInvestigador INT NOT NULL,

```

```

268     orcid VARCHAR(30) NOT NULL,
269     email VARCHAR(150) NOT NULL,
270     movil VARCHAR(15) NOT NULL,
271     idPersona INT NOT NULL,
272     PRIMARY KEY (idInvestigador),
273     FOREIGN KEY (idPersona) REFERENCES Persona(idPersona)
274 );
275
276 CREATE TABLE Proyecto
277 (
278     idProyecto INT NOT NULL,
279     nombre VARCHAR(250) NOT NULL,
280     inicio DATE NOT NULL,
281     final DATE NOT NULL,
282     idInvestigador INT NOT NULL,
283     PRIMARY KEY (idProyecto),
284     FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
285 );
286
287 CREATE TABLE Produccion
288 (
289     idProduccion INT NOT NULL,
290     tipo VARCHAR(60) NOT NULL,
291     titulo VARCHAR(250) NOT NULL,
292     anio INT NOT NULL,
293     idInvestigador INT NOT NULL,
294     PRIMARY KEY (idProduccion),
295     FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
296 );
297
298 CREATE TABLE Adscripcion
299 (
300     idAdscripcion INT NOT NULL,
301     instituto VARCHAR(80) NOT NULL,
302     area VARCHAR(150) NOT NULL,
303     nombramiento VARCHAR(10) NOT NULL,
304     fechaingreso DATETIME NOT NULL,
305     idInvestigador INT NOT NULL,
306     PRIMARY KEY (idAdscripcion),
307     FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
308 );
309
310 CREATE TABLE Formacion
311 (
312     idFormacion INT NOT NULL,
313     grado VARCHAR(18) NOT NULL,
314     institucion VARCHAR(70) NOT NULL,
315     nombre VARCHAR(120) NOT NULL,
316     fechatermino DATE NOT NULL,
317     idInvestigador INT NOT NULL,
318     PRIMARY KEY (idFormacion),
319     FOREIGN KEY (idInvestigador) REFERENCES Investigador(idInvestigador)
320 );
321
322 CREATE TABLE Asignatura
323 (
324     idAsignatura INT NOT NULL,
325     semestre INT NOT NULL,
326     nombre VARCHAR(120) NOT NULL,
327     idPrograma INT NOT NULL,
328     PRIMARY KEY (idAsignatura),
329     FOREIGN KEY (idPrograma) REFERENCES Programa(idPrograma)
330 );
331
332 CREATE TABLE IntegranteLinea
333 (
334     idIntegranteLinea INT NOT NULL,
335     vigente BOOLEAN NOT NULL,

```

```

336     inicio DATE NOT NULL,
337     termino DATE NOT NULL,
338     idLinea INT NOT NULL,
339     idIntegrante INT NOT NULL,
340     PRIMARY KEY (idIntegranteLinea),
341     FOREIGN KEY (idLinea) REFERENCES Linea(idLinea),
342     FOREIGN KEY (idIntegrante) REFERENCES Integrante(idIntegrante)
343 );
344
345 CREATE TABLE Curso
346 (
347     idCurso INT NOT NULL,
348     periodo VARCHAR(15) NOT NULL,
349     anio INT NOT NULL,
350     grupo INT NOT NULL,
351     idProfesor INT NOT NULL,
352     idAsignatura INT NOT NULL,
353     PRIMARY KEY (idCurso),
354     FOREIGN KEY (idProfesor) REFERENCES Profesor(idProfesor),
355     FOREIGN KEY (idAsignatura) REFERENCES Asignatura(idAsignatura)
356 );
357
358 CREATE TABLE AlumnoCurso
359 (
360     idAlumnoCurso INT NOT NULL,
361     calificacion FLOAT NOT NULL,
362     idCurso INT NOT NULL,
363     idAlumno INT NOT NULL,
364     PRIMARY KEY (idAlumnoCurso),
365     FOREIGN KEY (idCurso) REFERENCES Curso(idCurso),
366     FOREIGN KEY (idAlumno) REFERENCES Alumno(idAlumno)
367 );

```

Scripts de extracción de datos

```
1 USE Local_Profesor;
2 SELECT * FROM Programa
3 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Programa.txt'
4 FIELDS TERMINATED BY ','
5 OPTIONALLY ENCLOSED BY '"'
6 LINES TERMINATED BY '\n';
7
8 SELECT * FROM Alumno
9 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Alumno.txt'
10 FIELDS TERMINATED BY ','
11 OPTIONALLY ENCLOSED BY '"'
12 LINES TERMINATED BY '\n';
13
14 SELECT * FROM Profesor
15 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Profesor.txt'
16 FIELDS TERMINATED BY ','
17 OPTIONALLY ENCLOSED BY '"'
18 LINES TERMINATED BY '\n';
19
20 SELECT * FROM Asignatura
21 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Asignatura.txt'
22 FIELDS TERMINATED BY ','
23 OPTIONALLY ENCLOSED BY '"'
24 LINES TERMINATED BY '\n';
25
26 SELECT * FROM Curso
27 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Curso.txt'
28 FIELDS TERMINATED BY ','
29 OPTIONALLY ENCLOSED BY '"'
30 LINES TERMINATED BY '\n';
31
32 SELECT * FROM AlumnoCurso
33 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/AlumnoCurso.txt'
34 FIELDS TERMINATED BY ','
35 OPTIONALLY ENCLOSED BY '"'
36 LINES TERMINATED BY '\n';
37
38 -- Integrante --
39 USE Local_Integrante;
40 SELECT * FROM Persona
41 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Persona.txt'
42 FIELDS TERMINATED BY ','
43 OPTIONALLY ENCLOSED BY '"'
44 LINES TERMINATED BY '\n';
45
46 SELECT * FROM Integrante
47 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Integrante.txt'
48 FIELDS TERMINATED BY ','
49 OPTIONALLY ENCLOSED BY '"'
50 LINES TERMINATED BY '\n';
51
52 SELECT * FROM CuerpoAcademico
53 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/CuerpoAcademico.txt'
54 FIELDS TERMINATED BY ','
55 OPTIONALLY ENCLOSED BY '"'
56 LINES TERMINATED BY '\n';
57
58 SELECT * FROM Linea
59 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Linea.txt'
60 FIELDS TERMINATED BY ','
61 OPTIONALLY ENCLOSED BY '"'
62 LINES TERMINATED BY '\n';
63
64 SELECT * FROM IntegranteLinea
65 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/IntegranteLinea.txt'
66 FIELDS TERMINATED BY ','
```

```

67 OPTIONALLY ENCLOSED BY ''
68 LINES TERMINATED BY '\n';
69
70 -- Investigador --
71 USE Local_Investigador;
72 SELECT * FROM Investigador
73 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Investigador.txt'
74 FIELDS TERMINATED BY ','
75 OPTIONALLY ENCLOSED BY ''
76 LINES TERMINATED BY '\n';
77
78 SELECT * FROM Proyecto
79 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Proyecto.txt'
80 FIELDS TERMINATED BY ','
81 OPTIONALLY ENCLOSED BY ''
82 LINES TERMINATED BY '\n';
83
84 SELECT * FROM Produccion
85 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Produccion.txt'
86 FIELDS TERMINATED BY ','
87 OPTIONALLY ENCLOSED BY ''
88 LINES TERMINATED BY '\n';
89
90 SELECT * FROM Adscripcion
91 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Adscripcion.txt'
92 FIELDS TERMINATED BY ','
93 OPTIONALLY ENCLOSED BY ''
94 LINES TERMINATED BY '\n';
95
96 SELECT * FROM Formacion
97 INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Formacion.txt'
98 FIELDS TERMINATED BY ','
99 OPTIONALLY ENCLOSED BY ''
100 LINES TERMINATED BY '\n';

```

Scripts de carga de datos

```
1 USE Global_u;
2 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Persona.txt'
3 INTO TABLE Persona
4 FIELDS TERMINATED BY ','
5 OPTIONALLY ENCLOSED BY '"'
6 LINES TERMINATED BY '\n';
7
8 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Integrante.txt'
9 INTO TABLE Integrante
10 FIELDS TERMINATED BY ','
11 OPTIONALLY ENCLOSED BY '"'
12 LINES TERMINATED BY '\n';
13
14 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/CuerpoAcademico.txt'
15 INTO TABLE CuerpoAcademico
16 FIELDS TERMINATED BY ','
17 OPTIONALLY ENCLOSED BY '"'
18 LINES TERMINATED BY '\n';
19
20 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Linea.txt'
21 INTO TABLE Linea
22 FIELDS TERMINATED BY ','
23 OPTIONALLY ENCLOSED BY '"'
24 LINES TERMINATED BY '\n';
25
26 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/IntegranteLinea.txt'
27 INTO TABLE IntegranteLinea
28 FIELDS TERMINATED BY ','
29 OPTIONALLY ENCLOSED BY '"'
30 LINES TERMINATED BY '\n';
31
32
33 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Investigador.txt'
34 INTO TABLE Investigador
35 FIELDS TERMINATED BY ','
36 OPTIONALLY ENCLOSED BY '"'
37 LINES TERMINATED BY '\n';
38
39 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Proyecto.txt'
40 INTO TABLE Proyecto
41 FIELDS TERMINATED BY ','
42 OPTIONALLY ENCLOSED BY '"'
43 LINES TERMINATED BY '\n';
44
45 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Produccion.txt'
46 INTO TABLE Produccion
47 FIELDS TERMINATED BY ','
48 OPTIONALLY ENCLOSED BY '"'
49 LINES TERMINATED BY '\n';
50
51 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Adscripcion.txt'
52 INTO TABLE Adscripcion
53 FIELDS TERMINATED BY ','
54 OPTIONALLY ENCLOSED BY '"'
55 LINES TERMINATED BY '\n';
56
57 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Formacion.txt'
58 INTO TABLE Formacion
59 FIELDS TERMINATED BY ','
60 OPTIONALLY ENCLOSED BY '"'
61 LINES TERMINATED BY '\n';
62
63
64 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Programa.txt'
65 INTO TABLE Programa
66 FIELDS TERMINATED BY ','
```

```
67 OPTIONALLY ENCLOSED BY '"'
68 LINES TERMINATED BY '\n';
69
70 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Alumno.txt'
71 INTO TABLE Alumno
72 FIELDS TERMINATED BY ','
73 OPTIONALLY ENCLOSED BY '"'
74 LINES TERMINATED BY '\n';
75
76 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Profesor.txt'
77 INTO TABLE Profesor
78 FIELDS TERMINATED BY ','
79 OPTIONALLY ENCLOSED BY '"'
80 LINES TERMINATED BY '\n';
81
82 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Asignatura.txt'
83 INTO TABLE Asignatura
84 FIELDS TERMINATED BY ','
85 OPTIONALLY ENCLOSED BY '"'
86 LINES TERMINATED BY '\n';
87
88 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Curso.txt'
89 INTO TABLE Curso
90 FIELDS TERMINATED BY ','
91 OPTIONALLY ENCLOSED BY '"'
92 LINES TERMINATED BY '\n';
93
94 LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/AlumnoCurso.txt'
95 INTO TABLE AlumnoCurso
96 FIELDS TERMINATED BY ','
97 OPTIONALLY ENCLOSED BY '"'
98 LINES TERMINATED BY '\n';
99 LINES TERMINATED BY '\n';
```


Script Consulta como ejemplo

```
1  -- Consulta: Profesores que tambi n son investigadores y su producci n acad mica
2  SELECT
3      p.paterno,
4      p.materno,
5      p.nombre,
6      prof.email AS email_profesor,
7      inv.orcid AS id_orcid,
8      COUNT(pr.idProduccion) AS total_publicaciones
9  FROM
10     Global_u.Persona p
11  JOIN
12     Global_u.Profesor prof ON p.idPersona = prof.idPersona
13  JOIN
14     Global_u.Investigador inv ON p.idPersona = inv.idPersona
15  LEFT JOIN
16     Global_u.Produccion pr ON inv.idInvestigador = pr.idInvestigador
17  GROUP BY
18     p.idPersona, p.paterno, p.materno, p.nombre, prof.email, inv.orcid
19  ORDER BY
20     p.paterno, p.materno;
```

	paterno	materno	nombre	email_profesor	id_orcid	total_publicaciones
►	Acosta	Maldonado	Laura Isabel	maria.rangel@universidad.edu	0000-0050-1234-5678	1
	Aguilar	Rosas	Manuel Salvador	hector.villarreal@universidad.edu	0000-0037-8901-2345	1
	Aguirre	Franco	María Teresa	laura.aguirre@universidad.edu	0000-0072-3456-7890	1
	Andrade	Cárdenas	Verónica Elizabeth	maria.lozano@universidad.edu	0000-0074-5678-9012	1
	Barrera	Tapia	Silvia Fernanda	guadalupe.rivas@universidad.edu	0000-0070-1234-5678	1
	Bautista	Zamora	Jorge Octavio	juan.fonseca@universidad.edu	0000-0097-8901-2345	1
	Bernal	Carmona	Francisco Javier	javier.enriquez@universidad.edu	0000-0087-8901-2345	1
	Camacho	Castañeda	Ricardo Ernesto	juan.zamora@universidad.edu	0000-0081-2345-6789	1
	Campos	Santos	María José	carmen.rios@universidad.edu	0000-0028-9012-3456	1
	Cárdenas	Márquez	Luis Felipe	jorge.carmona@universidad.edu	0000-0077-8901-2345	1
	Carmona	Corona	Ana Lilia	ana.palacios@universidad.edu	0000-0090-1234-5678	1

5. Conclusiones

La estrategia Bottom-Up demostró ser altamente efectiva para el diseño de la base de datos, ya que permitió construir el esquema de manera progresiva, asegurando que cada tabla y relación estuviera bien definida antes de integrarla a estructuras más complejas. La exportación de datos a archivos .txt y su posterior carga mediante comandos SQL garantizaron una migración eficiente, manteniendo la consistencia y estructura relacional.

Esta práctica reforzó cómo un enfoque estructurado y modular en el diseño de bases de datos no solo optimiza el rendimiento, sino que también facilita futuras actualizaciones y escalabilidad del sistema. Como trabajo futuro, se podría explorar la automatización de estos procesos mediante scripts o la implementación de herramientas ETL para gestionar grandes volúmenes de datos.

Referencias Bibliográficas

References

- [1] Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley.
- [2] Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems*. Pearson.
- [3] Date, C. J. (2004). *An Introduction to Database Systems*. Addison-Wesley.
- [4] Inmon, W. H. (2005). *Building the Data Warehouse*. Wiley.
- [5] Grabowska, S.; Saniuk, S. (2022). Business models in the industry 4.0 environment—results of web of science bibliometric analysis. *J. Open Innov. Technol. Mark. Complex*, 8(1), 19.
- [6] Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th ed.). Pearson. (Capítulo 10: Diseño bottom-up).