

Unidad 2

Diseño físico e implementación de
bases de datos

UTPL
UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

Agenda

- Introducción al Diseño Físico
- Traducción del modelo lógico a lenguaje del SGBD
- Organización física de la base de datos
- Implementación de la base de datos
- Administración de la base de datos



Introducción al Diseño Físico

- En el Diseño Físico se establecen las **especificaciones de implementación** de la base de datos en un entorno específico:
 - SGBD a usar
 - Sistema Operativo
 - Requerimientos de almacenamiento secundario
 - Configuraciones para acceso eficiente
 - Vistas de usuario
 - Mecanismos de seguridad, etc.

Introducción al Diseño Físico

- Diseño Lógico se preocupa del “Que” el diseño físico del “Como”
- El Diseño Físico exige conocer a fondo las capacidades y prestaciones del SGBD elegido.
- El Diseño Físico no es aislado, puede implicar volver atrás.
- En el Diseño Físico aparece el rol DBA



Introducción al Diseño Físico

Ciclo de vida de la base de datos

- Estudio inicial
 - Definición de necesidades
 - Recopilación y análisis de requisitos
 - Diseño de la base de datos
 - Diseño Conceptual
 - Diseño Lógico
 - **Diseño Físico**
 - Implementación
 - Conversión y carga de datos
 - Pruebas
 - Implantación (puesta en operación)
 - Mantenimiento operativo
-
- Administrador de los datos**
- Administrador de la base de datos (DBA)**

Introducción al Diseño Físico

Proceso del Diseño Físico

- El proceso que conlleva el diseño físico puede variar dependiendo de los autores y de las necesidades del proyecto.
- En general implica los siguientes pasos:
 1. Traducir el modelo lógico a lenguaje del SGBD
 2. Definir la organización física de la base de datos
 3. Identificar necesidades de indexación
 4. Definir mecanismos y restricciones de seguridad
 5. Establecer protocolo de monitoreo

Introducción al Diseño Físico

Proceso del Diseño Físico

Paso 1: Traducir el modelo lógico a lenguaje del SGBD

Script de implementación de la base de datos

```
1  -- Versión del servidor:      5.7.24 - MySQL Community Server (GPL)
2  --
3  --
4  --
5  /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@CHARACTER_SET_CLIENT */;
6  /*!40101 SET NAMES utf8 */;
7  /*!40101 SET @OLD_SQL_MODE=@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
8  --
9  --
10 -- Volcando estructura de base de datos para ecolac
11 CREATE DATABASE IF NOT EXISTS `ecolac` /*!40100 DEFAULT CHARACTER SET utf8mb4 */;
12 USE `ecolac`;
13 --
14 -- Volcando estructura para tabla ecolac.categories
15 CREATE TABLE IF NOT EXISTS `categories` (
16     `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
17     `nombre` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
18     `slug` varchar(50) COLLATE utf8mb4_unicode_ci NOT NULL,
19     `descripcion` varchar(255) COLLATE utf8mb4_unicode_ci DEFAULT NULL,
20     `created_at` timestamp NULL DEFAULT NULL,
21     `updated_at` timestamp NULL DEFAULT NULL,
22     PRIMARY KEY (`id`),
23     UNIQUE KEY `categories_nombre_unique` (`nombre`),
24     UNIQUE KEY `categories_slug_unique` (`slug`)
25 ) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4;
26 --
27 --
28 -- Volcando estructura para tabla ecolac.ciudades
29 CREATE TABLE IF NOT EXISTS `ciudades` (
30     `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
31     `nombre` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
32     `slug` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
33     PRIMARY KEY (`id`),
34     UNIQUE KEY `ciudades_nombre_unique` (`nombre`),
35     UNIQUE KEY `ciudades_slug_unique` (`slug`)
36 ) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4;
```

Paso 2: Definir la organización física de la base de datos

Paso 3: Identificar necesidades de indexación

Paso 4: Definir mecanismos y restricciones de seguridad

Paso 5: Establecer protocolo de monitoreo y otras especificaciones de implementación

Traducción modelo lógico a SGBD

- Requiere conocimiento de la **organización y estructuración lógica** del SGBD elegido
 - Estructuras lógicas de almacenamiento de los datos
 - Restricciones que se pueden implementar
- Requiere conocimiento del lenguaje de implementación de la base de datos (**DDL** de SQL en bases de datos relacionales)

Traducción modelo lógico a SGBD

- Se parte del modelo lógico (diagrama + diccionario de datos)
- Proceso
 - Evaluar posibles limitaciones de implementación en el SGBD elegido.
 - Traducir relaciones base
 - Traducir integridad referencial
 - Traducir otras restricciones de integridad y de dominio

Traducción modelo lógico a SGBD

- Mecanismos para implementar el modelo lógico en el SGBD elegido.
 - Todo SGBD provee de un lenguaje para implementar la base de datos.
 - En un SGBD **Relacional** éste lenguaje se denomina **DDL** (Data Definition Language) que es parte de SQL.
 - DDL permite definir la mayoría de las especificaciones del modelo, pero no todas.
 - **Las restricciones que no permita implementar el SGBD se las debe controlar a nivel de aplicación o con triggers.**

Traducción modelo lógico a SGBD

- Usualmente la mayoría de los SGBD relacionales solo permiten implementar las siguientes 5 restricciones de integridad mediante DDL:
 - Integridad de entidades (PRIMARY KEY)
 - Integridad referencial (FOREIGN KEY)
 - Claves alternativas (UNIQUE KEY)
 - Campos obligatorios (NOT NULL)
 - Validación de dominio (CHECK)

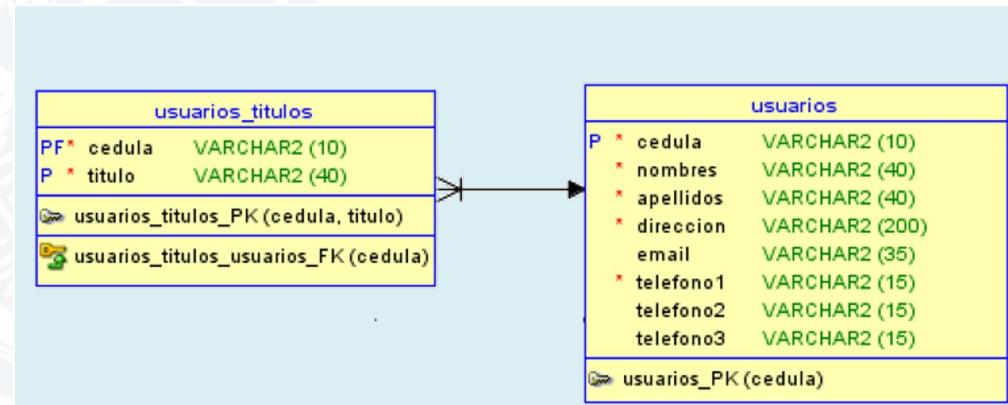
Traducción modelo lógico a SGBD

- Ejemplos de limitaciones que puede tener un SGBD
 - En MySQL la implementación de integridad referencial solo es permitida en tablas tipo InnoDB



Traducción modelo lógico a SGBD

- Restricciones que **no se pueden implementar** con DDL:
 - **Campos calculados:** la posibilidad de que el recálculo se haga automático.
 - **Cardinalidad mínima 1 del extremo muchos en relaciones 1:N:** No se puede agregar registros a dos tablas en una sola operación.
 - **Cardinalidad máxima distinta a N en relaciones 1:N:** No se puede controlar el máximo asociaciones mediante una llave foránea.

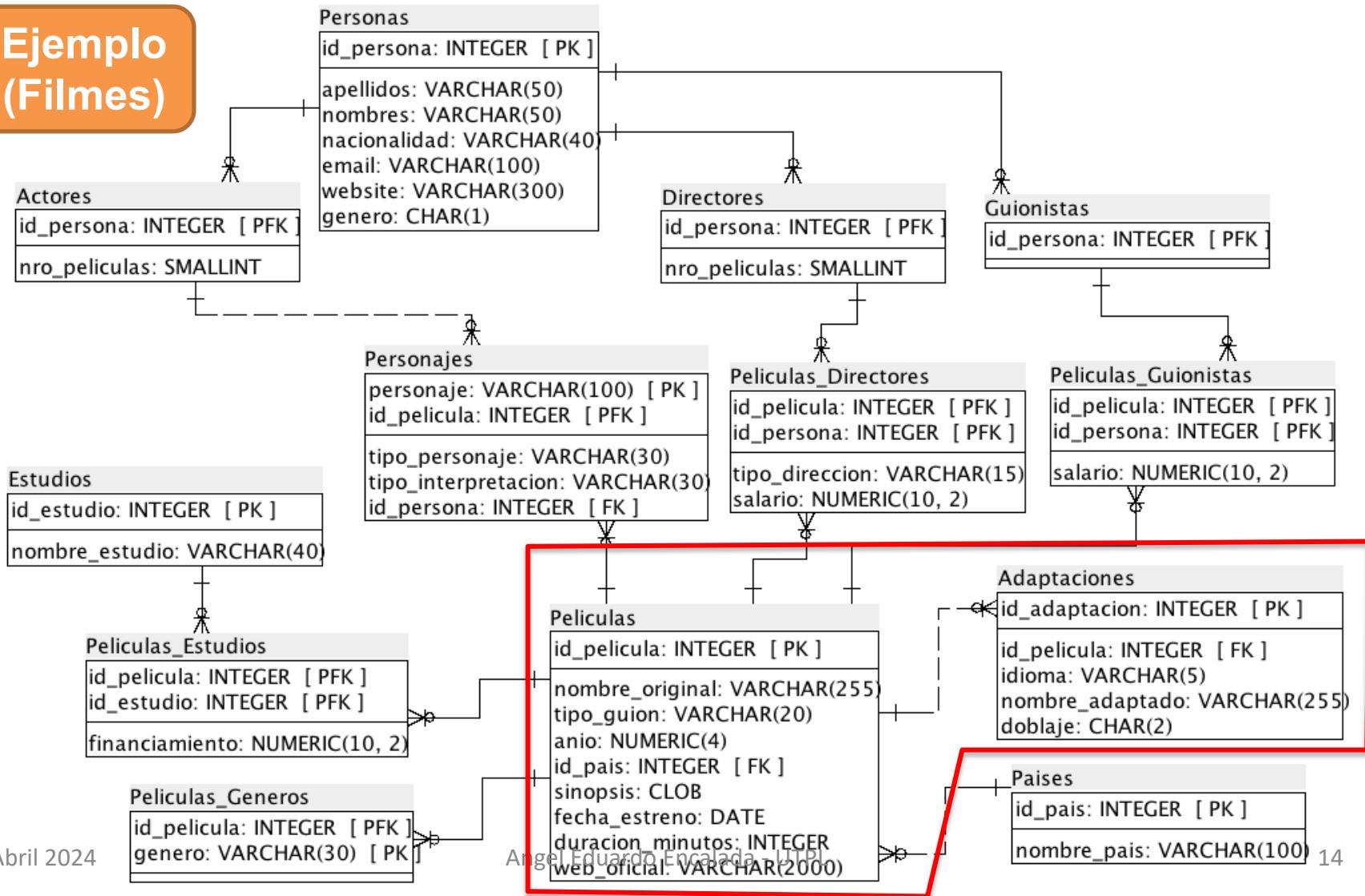


Un usuario debe tener al menos un título.

Un usuario no puede tener más de 3 títulos

Traducción modelo lógico a SGBD

Ejemplo (Filmes)



Traducción modelo lógico a SGBD

DDL
Ejemplo
(Filmes)

```
CREATE TABLE Peliculas (
    id_pelicula INTEGER NOT NULL,
    nombre_original VARCHAR(255) NOT NULL,
    tipo_guion VARCHAR(20) NOT NULL,
    anio NUMERIC(4) NOT NULL,
    id_pais INTEGER NOT NULL,
    sinopsis CLOB NOT NULL,
    fecha_estreno DATE,
    duracion_minutos INTEGER NOT NULL,
    web_oficial VARCHAR(2000),
    CONSTRAINT Peliculas_pk PRIMARY KEY (id_pelicula));
```

```
CREATE TABLE Adaptaciones (
    id_adaptacion INTEGER NOT NULL,
    id_pelicula INTEGER NOT NULL,
    idioma VARCHAR(5) NOT NULL,
    nombre_adaptado VARCHAR(255),
    doblaje CHAR(2),
    CONSTRAINT adaptaciones_pk PRIMARY KEY (id_adaptacion));
```

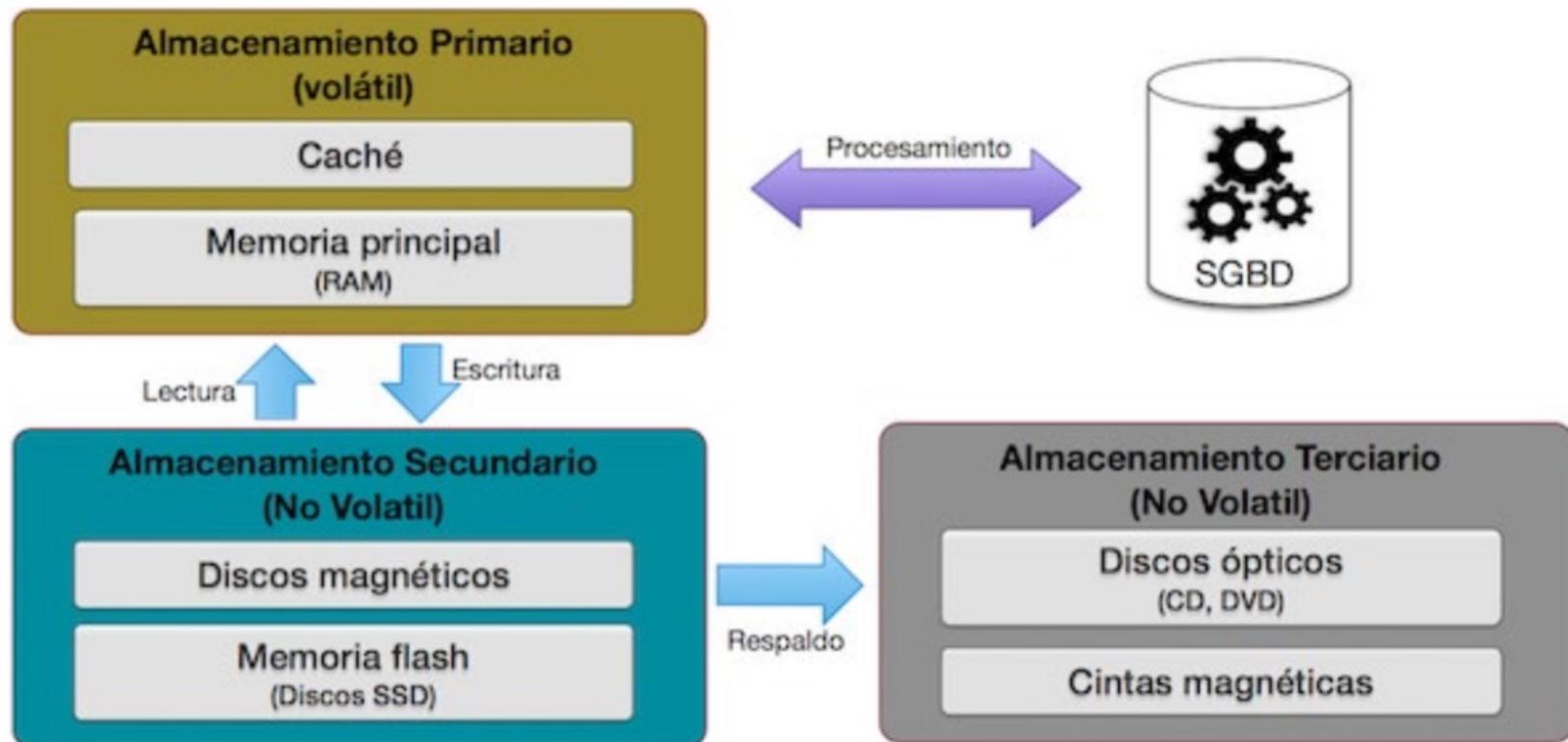
```
ALTER TABLE Adaptaciones ADD CONSTRAINT Peliculas_Adaptaciones_fk
FOREIGN KEY (id_pelicula) REFERENCES Peliculas (id_pelicula)
ON DELETE NO ACTION ON UPDATE NO ACTION;
```

Organización física de la Base de Datos

- Busca identificar las especificaciones a nivel de almacenamiento físico de los datos.
- Se debe determinar sobre todo:
 - El espacio requerido en almacenamiento secundario.
 - La distribución física de los datos.
- Requiere conocimiento de la **organización y estructuración física** que maneja el SGBD e incluso el sistema operativo.
- Una adecuada organización física de los datos puede ayudar a mejorar el rendimiento de la base de datos.

Organización física de la Base de Datos

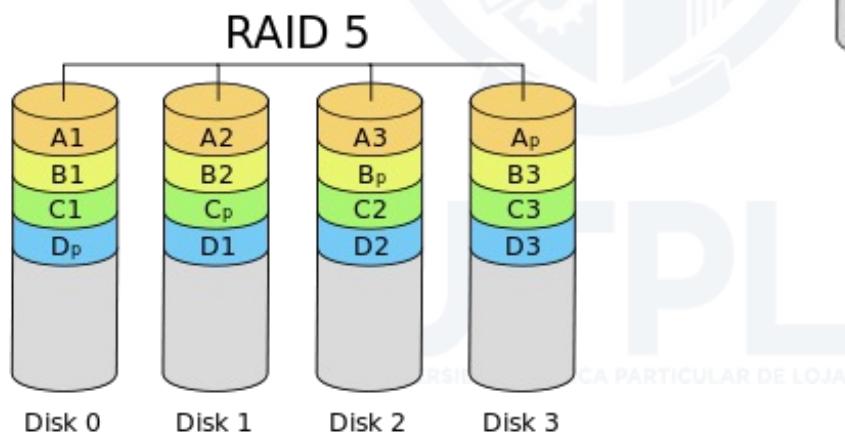
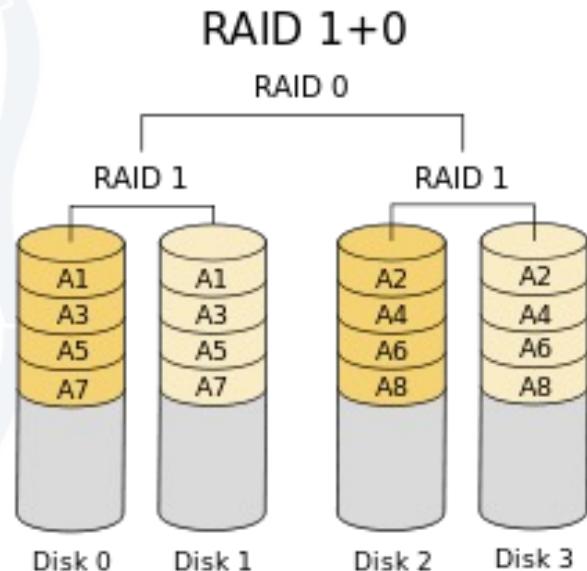
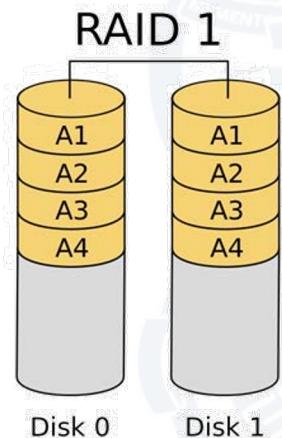
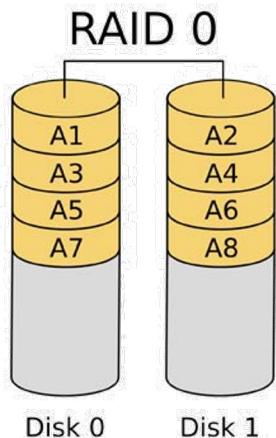
Medios de Almacenamiento



Organización física de la Base de Datos

RAID

(Redundant Array of Independent Disks)



Análisis transaccional

- Para establecer las especificaciones de organización física de los datos se puede partir de un **Análisis Transaccional**
- El objetivo del análisis transaccional es identificar:
 - **La relaciones (tablas) críticas, en base al índice de operaciones de consulta y actualización.**
 - **El espacio físico en disco requerido**
 - **La distribución física de los datos**

Análisis transaccional

- Busca cuantificar la carga y comportamiento que tendrá la base de datos, a fin de:
 - Identificar tablas críticas
 - Identificar transacciones críticas
- No es una ciencia exacta
- Requiere lista de transacciones que se ejecutarán en la base de datos.



Análisis transaccional

Ejemplo lista de transacciones

Ejemplo
(Filmes)

TRANSACCIONES		
	Transacción	Tipo
1	Registrar una nueva película	Actualizacion
2	Gestionar Actores	Actualizacion
3	Gestionar Directores	Actualizacion
4	Gestionar Guionistas	Actualizacion
5	Actualizar Informacion de Películas	Actualizacion
6	Mostrar ficha informativa de una película	Consulta
7	Mostrar Personajes Interpretados por un actor	Consulta
8	Numero de películas dirigidas por un director	Consulta
9	Listar películas adaptadas a un idioma (ej ES)	Consulta
10	Listar Películas según el género	Consulta
11	Listar Películas con mayor presupuesto	Consulta
12	Listar actores mejor pagados	Consulta

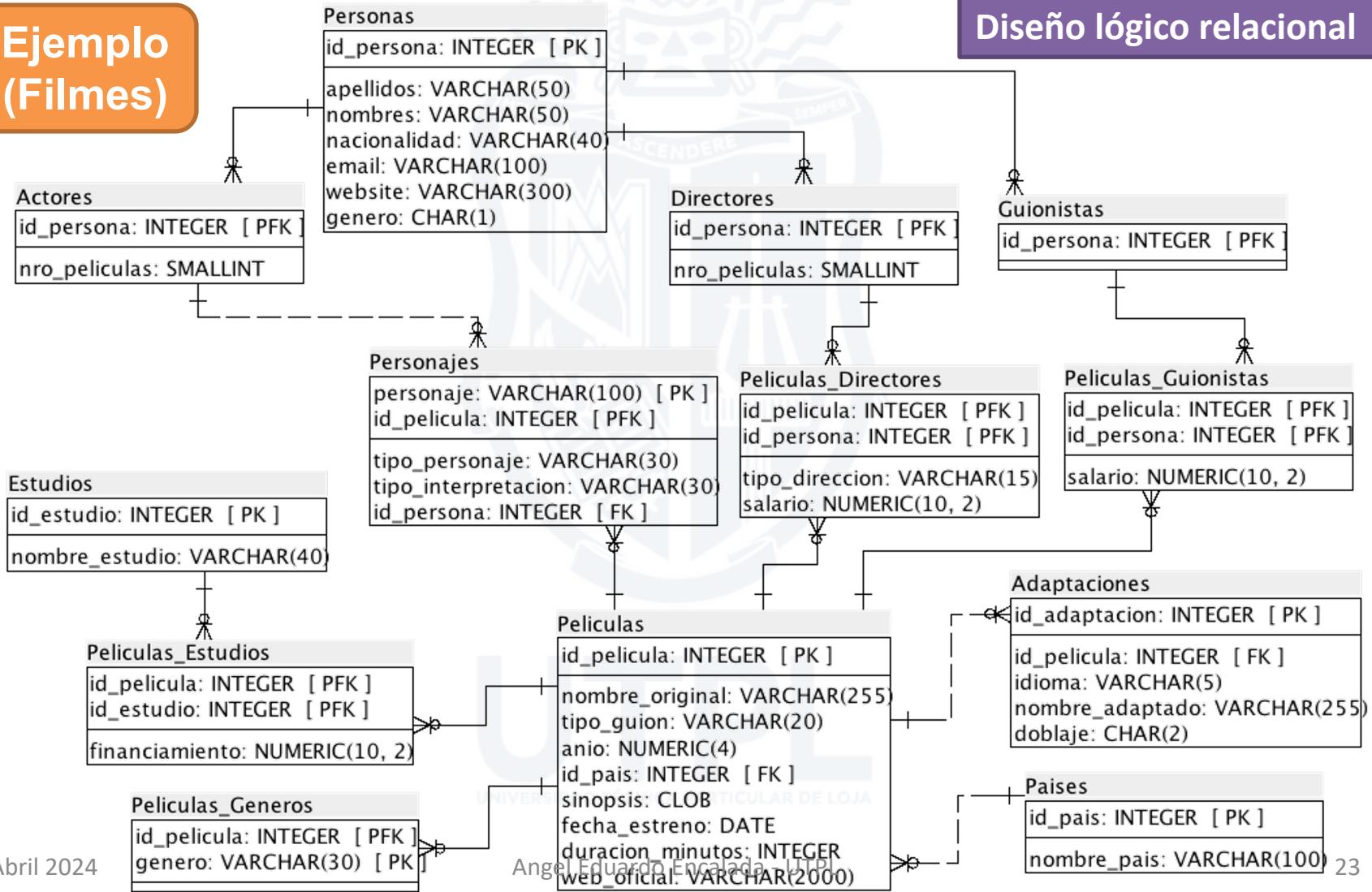
Análisis transaccional

- Por cada transacción se identifican las tablas que serán accedidas sea para operaciones de consulta o de actualización.
- Luego por cada tabla se contabiliza el número de accesos, con base en las transacciones que la usan y la frecuencia de esas transacciones.
- Igualmente por cada transacción se determina la cantidad de tablas que requiere acceder

Análisis transaccional

Ejemplo
(Filmes)

Diseño lógico relacional



Análisis transaccional

Matriz cruzada de transacciones y relaciones

Ejemplo
(Filmes)

Transacción	Matriz cruzada transacciones/relaciones														# Tablas por transacción
	TABLAS														
	Personas	Actores	Directores	Guionistas	Películas	Personajes	Películas_Directores	Películas_Guionistas	Estudios	Películas_Estudios	Películas_Generos	Adaptaciones	Países		
Registrar una nueva película	X	X	X	X	X	X	X	X	X	X	X	X	X	X	13
Gestionar Actores	X	X													2
Gestionar Directores	X		X												2
Gestionar Guionistas	X			X											2
Actualizar Informacion de Películas	X	X	X	X	X	X	X	X	X	X	X	X	X	X	13
Mostrar ficha informativa de una película	X	X	X	X	X	X	X	X	X	X	X	X	X	X	13
Mostrar Personajes Interpretados por un actor	X	X			X	X									4
Numero de películas dirigidas por un director	X		X												2
Listar películas adaptadas a un idioma (ej ES)					X							X			2
Listar Películas según el género					X						X				2
Listar Películas con mayor presupuesto					X					X					2
Listar actores mejor pagados	X	X			X										3
# Transacciones x tabla	9	6	5	4	8	4	3	3	3	4	4	4	4	3	

Proceso Diseño Físico - Paso 2: Definir la organización física de la base de datos

Análisis transaccional detallado. Considerando la frecuencia de ejecución de cada transacción.

C=Consulta
A=Actualización

TRANSACCIONES	Freq	Matriz cruzada transacciones/relaciones													# Tablas por transacción
		TABLAS													
		personas	actores	directores	guionistas	películas	personajes	películas_directores	películas_guionistas	estudios	películas_estudios	películas_generos	adadaptaciones	paises	
Registrar una nueva película	200	C	CA	CA	C	A	A	A	A	C	A	A	A	C	13
Gestionar actores	4	CA	CA												2
Gestionar directores	2	CA		CA											2
Gestionar guionistas	3	CA			CA										2
Actualizar Información de Películas	20	C	CA	CA	C	CA	CA	CA	CA	C	CA	CA	CA	C	13
Mostrar ficha informativa de una película	35000	C	C	C	C	C	C	C	C	C	C	C	C	C	13
Mostrar Personajes Interpretados por un actor	3000	C	C			C	C								4
Obtener número de películas dirigidas por un director	100	C		C											2
Listar películas adaptadas a un idioma (ej. ES)	15000					C						C			2
Listar Películas según el género	30000					C						C			2
Listar Películas con mayor presupuesto	50					C						C			2
Listar directores mejor pagados	800	C		C				C							3
# Accesos de consulta >>>		39129	38224	36122	35223	83070	38020	35820	35020	35220	35070	65020	50020	35220	
# Accesos de actualización >>>		9	224	222	3	220	220	220	220	0	220	220	220	0	
# Total accesos >>>		39138	38448	36344	35226	83290	38240	36040	35240	35220	35290	65240	50240	35220	

Análisis transaccional

Resultados del análisis transaccional

Ejemplo
(Filmes)

Resultados:

Se trata de una base de datos predominantemente de consulta

- Tablas más accedidas:
 - PELICULAS
 - PELICULAS_GENEROS
 - ADAPTACIONES
- Tablas más consultadas
 - PELICULAS
 - PELICULAS_GENEROS
 - ADAPTACIONES
- En cuanto a actualizaciones la incidencia no es alta, y no existen tablas donde predomine esta operación.

Organización física de la Base de Datos

Espacio requerido en almacenamiento secundario

- Se calcula con base en:
 - Tamaño de registro de cada tabla
 - Volumen de carga inicial de datos (# de tuplas por tabla)
 - Frecuencia de inserciones diarias
 - Considerar espacio adicional (20%)
 - Proyección a 5 años mínimo
 - Si una tabla tiene asociado archivos digitales a las tuplas (jpg, pdf, doc ...), se debe incluir en el tamaño del registro.
 - Dependiendo del SGBD se deberá prever espacio para otros archivos que no son de datos (Ej: REDOLOGS en Oracle)

Organización física de la Base de Datos

Espacio requerido en almacenamiento secundario

**Ejemplo
(Filmes)**

Tipos de datos Oracle y su tamaño en bytes		
Tipo de dato	Tamaño en bytes	Observaciones
CHAR VARCHAR CLOB	1 byte por cada carácter	<p>Por ejemplo:</p> <ul style="list-style-type: none"> - CHAR (25) -> Significa que ocupará exactamente 25 Bytes - VARCHAR(25) -> Significa que ocupará un máximo de 25 Bytes (Para cálculos de espacio tomar en cuenta el máximo) - Para el caso de CLOB la cantidad de caracteres que se pueden almacenar es ilimitado (hasta 4GB). Se debe establecer un tamaño estimado promedio para los cálculos de espacio requerido.
INTEGER	4 bytes	
SMALLINT	2 bytes	
DATE	7bytes	
NUMERIC	de 1 a 22 bytes	Depende de la longitud y precisión que se establezca. Lo más común es que ocupe entre 6 y 8 bytes
ROWID	10 bytes	Es la dirección física de cada tupla

Organización física de la Base de Datos

Espacio requerido en almacenamiento secundario

**Ejemplo
(Filmes)**

Incluido el
ROWID

Relaciones	Tamaño aproximado de cada registro (en Bytes)	Carga Inicial		Carga Diaria	Proyección 5 años		Total a 5 años	
		Registros	Bytes	Registros	Registros	Bytes	Bytes	MB
Personas	555	21,000	11,655,000	2.1	3,780	2,097,900	13,752,900	13.12
Actores	16	15,000	240,000	1.5	2,700	43,200	283,200	0.27
Directores	16	3,000	48,000	0.3	540	8,640	56,640	0.05
Guionistas	14	3,000	42,000	0.3	540	7,560	49,560	0.05
Peliculas	6,510	250,000	1,627,500,000	30.0	54,000	351,540,000	1,979,040,000	1,887.36
Personajes	178	3,750,000	667,500,000	450.0	810,000	144,180,000	811,680,000	774.08
Peliculas_Directores	41	1,000,000	41,000,000	120.0	216,000	8,856,000	49,856,000	47.55
Películas_Guionistas	26	375,000	9,750,000	45.0	81,000	2,106,000	11,856,000	11.31
Estudios	54	500	27,000	0.0	54	2,916	29,916	0.03
Peliculas_Estudios	26	500,000	13,000,000	60.0	108,000	2,808,000	15,808,000	15.08
Peliculas_Generos	44	500,000	22,000,000	60.0	108,000	4,752,000	26,752,000	25.51
Adaptaciones	280	1,250,000	350,000,000	150.0	270,000	75,600,000	425,600,000	405.88
Paises	114	200	22,800	0.0	0	0	22,800	0.02
TOTAL			2,742,784,800			592,002,216	3,334,787,016	3,180.30

Organización física de la Base de Datos

Espacio requerido en almacenamiento secundario

**Ejemplo
(Filmes)**

Incluido el
ROWID

```

Películas
id_película: INTEGER [ PK ]
nombre_original: VARCHAR(255)
tipo_guion: VARCHAR(20)
año: NUMERIC(4)
id_pais: INTEGER [ FK ]
sinopsis: CLOB
fecha_estreno: DATE
duración_minutos: INTEGER
web_oficial: VARCHAR(2000)

```

Relaciones	Tamaño aproximado de cada registro (en Bytes)	Carga Inicial		Carga Diaria	Proyección 5 años		Total a 5 años	
		Registros	Bytes	Registros	Registros	Bytes	Bytes	MB
Personas	555	21,000	11,655,000	2.1	3,780	2,097,900	13,752,900	13.12
Actores	16	15,000	240,000	1.5	2,700	43,200	283,200	0.27
Directores	16	3,000	48,000	0.3	540	8,640	56,640	0.05
Guionistas	14	3,000	42,000	0.3	540	7,560	49,560	0.05
Películas	6,510	250,000	1,627,500,000	30.0	54,000	351,540,000	1,979,040,000	1,887.36
Personajes	178	3,750,000	667,500,000	450.0	810,000	144,180,000	811,680,000	774.08
Películas_Directores	41	1,000,000	41,000,000	120.0	216,000	8,856,000	49,856,000	47.55
estas	26	375,000	9,750,000	45.0	81,000	2,106,000	11,856,000	11.31
os	54	500	27,000	0.0	54	2,916	29,916	0.03
os	26	500,000	13,000,000	60.0	108,000	2,808,000	15,808,000	15.08
os	44	500,000	22,000,000	60.0	108,000	4,752,000	26,752,000	25.51
	280	1,250,000	350,000,000	150.0	270,000	75,600,000	425,600,000	405.88
	114	200	22,800	0.0	0	0	22,800	0.02
			2,742,784,800			592,002,216	3,334,787,016	3,180.30

Organización física de la Base de Datos

Espacio requerido en almacenamiento secundario

Ejemplo
(Filmes)

TOTALES PROYECTADO A 5 AÑOS (en Mbytes)	
Espacio tablas	3,180.30
+20% (Actualizaciones y Metadata)	636.06
Total TABLAS	3,816.36
Espacio Adicional (archivos)	0.00
Total TABLAS + OTROS	3,816.36
+50% (Índices & Otros)	1,908.18
TOTAL	5,724.54

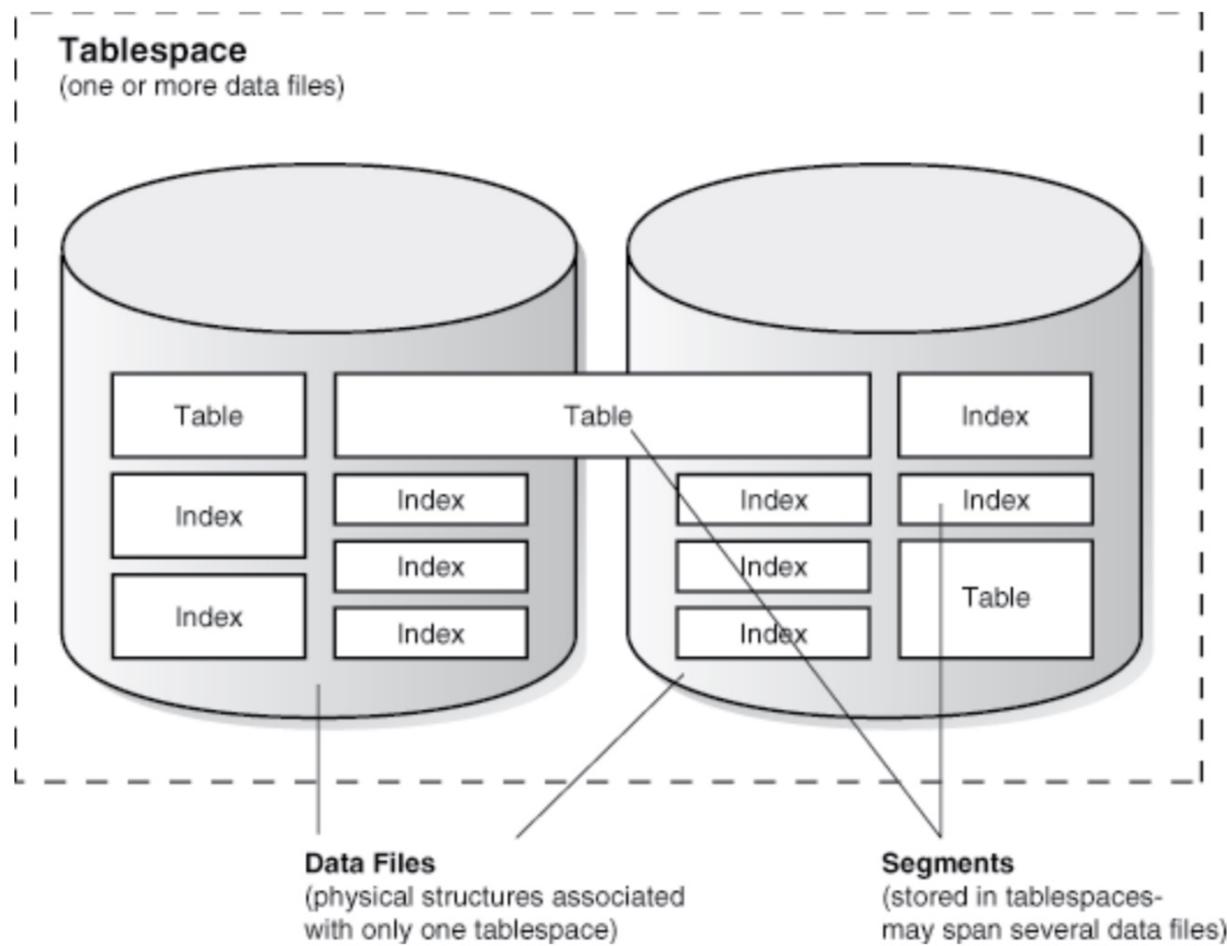
Organización física de la Base de Datos

Distribución física de los datos en disco

- Dependiendo de las facilidades del SGBD, se puede personalizar la distribución de los datos en disco.
- El objetivo es facilitar al motor el proceso de operaciones I/O. Es decir, las transferencias de bloques de datos entre almacenamiento primario y almacenamiento secundario.
- Por ejemplo: tablas que se acceden de manera concurrente, se las puede almacenar en ubicaciones distintas.

Organización física de la Base de Datos

Estructuras de almacenamiento físico en Oracle



Organización física de la Base de Datos

- La organización física de los datos se establece de manera específica de acuerdo al motor seleccionado. Por ejemplo, en el caso de Oracle:
 - Nombre, tamaño y ubicación de los tablespaces.
 - Distribución de tablas e índices en los tablespaces.
- Se debe actualizar el script DDL para incluir la organización física establecida.
- Adicionalmente, dentro de la organización física también se suele establecer la parametrización requerida en el SGBD, en cuanto a:
 - Tamaño memoria RAM reservada para el motor
 - Distribución de memoria entre los procesos del SGBD
 - Número máximo de sesiones concurrentes
 - Número de procesos de usuario
 - Configuración regional, etc.

Revisar:

- Libro de texto (Coronel et al., 2011): apartados
 - 7.2: Comandos para definición de datos.
 - 9.7: Diseño físico.
- Video (Encalada, 2016c) (<https://rebrand.ly/youtu3506>)
 - Bases de datos: Traducir modelo lógico a SGBD
- REA (Cabré et al., 2015) (<https://rebrand.ly/70025r4>): hasta el apartado 5
 - Diseño físico de bases de datos



Implementación de la base de datos

Ciclo de vida de la base de datos

- Estudio inicial
 - Definición de necesidades
 - Recopilación y análisis de requisitos
- Diseño de la base de datos
 - Diseño Conceptual
 - Diseño Lógico
 - Diseño Físico
- **Implementación y carga**
- Pruebas
- Implementación (puesta en operación)
- Mantenimiento operativo

**Administrador
de los datos**

**Administrador
de la base de
datos (DBA)**



Implementación de la base de datos

- Concluido el diseño físico, con el script DDL completo y con todas las especificaciones de configuración del SGBD, la implementación se reduce a:
 - Preparar el equipo servidor físico o virtual
 - Instalar y configurar el SGBD
 - Instalar software complementario
 - Crear el esquema de la base de datos
 - Crear los objetos de datos (tablas, vistas, índices, etc.).
 - Implementar los controles de seguridad (roles, perfiles, usuarios, permisos, etc.)

Script DDL

Carga inicial

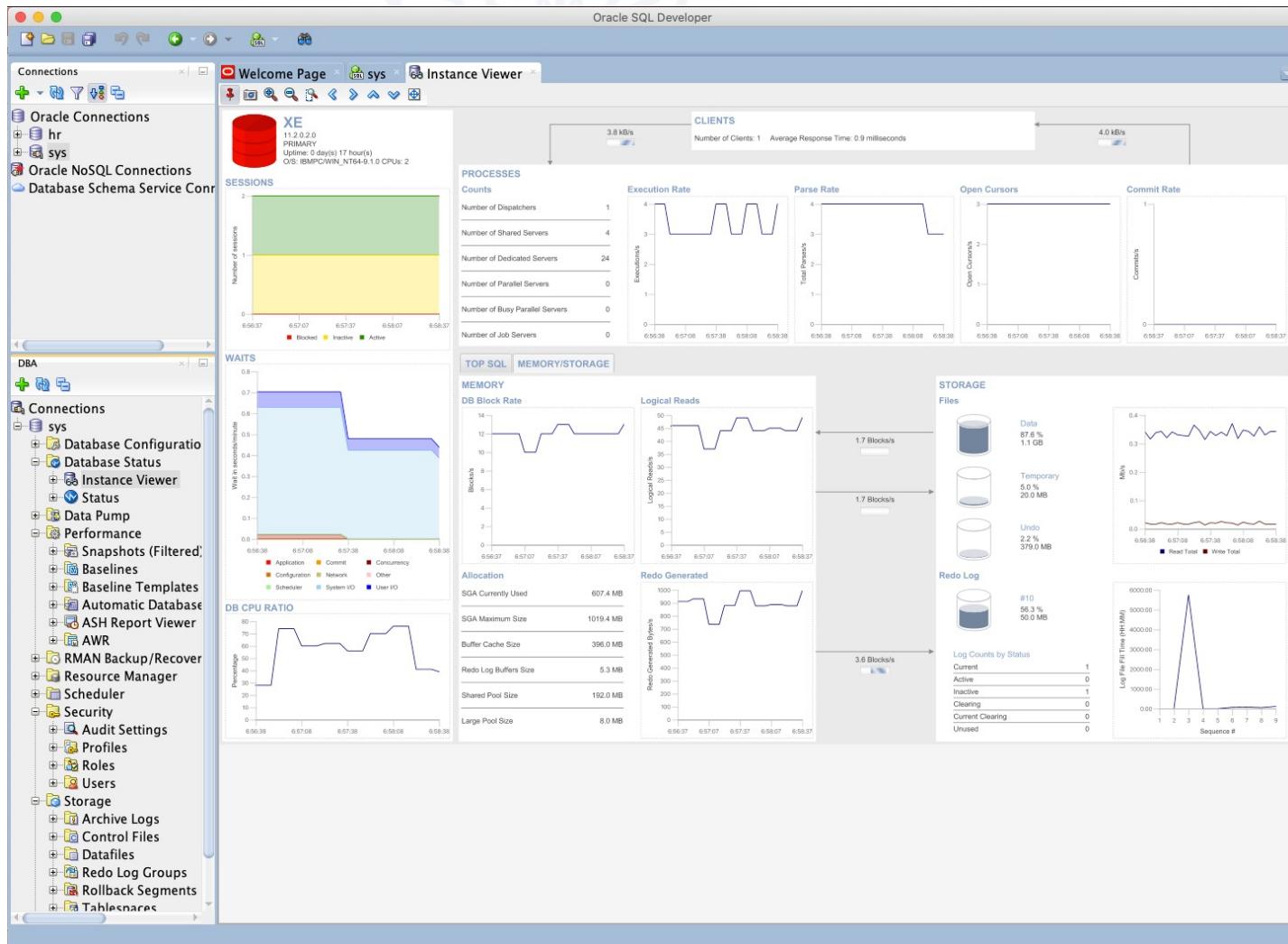
- Agregación de los datos con los que la base de datos entrará en producción.
- Puede ser una carga simple (catálogos, parámetros), o compleja (migrar los datos de una base anterior).
- El proceso de carga conlleva:
 - Identificar datos de partida
 - Seleccionar fuentes
 - Preparar los datos a cargar
 - Cargar los datos

Monitorización de la base de datos

- Realizar un seguimiento de la operación de la base de datos.
- Depende del SGBD: en la mayoría es posible activar utilidades de auditoría o instalar herramientas de monitoreo.
- Se monitorea principalmente: memoria, procesador y disco.
- Si se detectan problemas de rendimiento y ralentización de operaciones se deben realizar los ajustes que correspondan (afinamiento de la base de datos)

Monitorización de la base de datos

Ejemplo
monitoreo
Oracle con
SQL
Developer



Monitorización de la base de datos

Ejemplo
monitoreo
Oracle con
Oracle
Enterprise
Manager

