



西安电子科技大学
XIDIAN UNIVERSITY

12

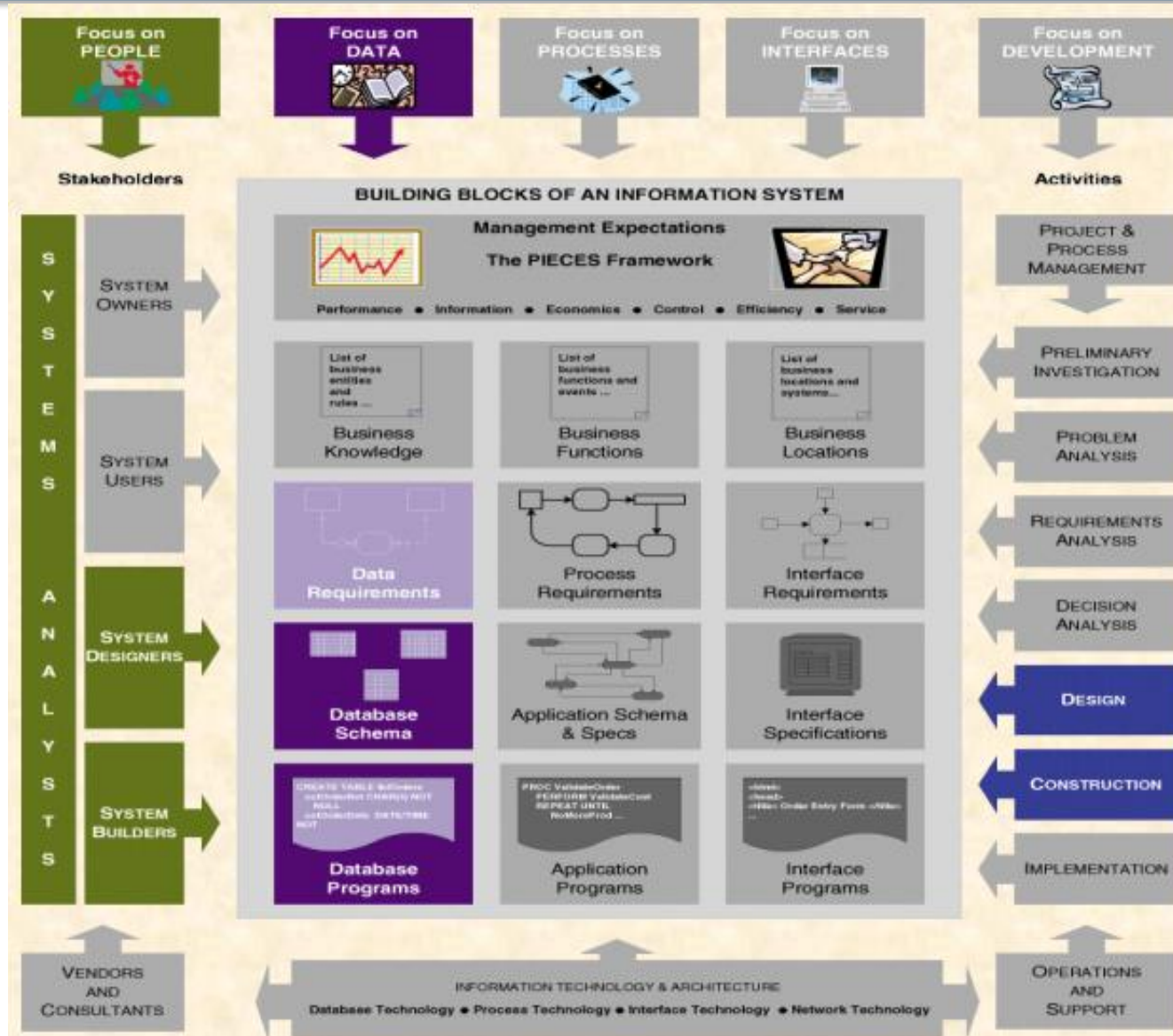
系统分析与设计 (SYSTEM ANALYSIS AND DESIGN)

Database Design

Content Structure

- ❁ Conventional Files versus the Database
- ❁ Database Concepts for the Systems Analyst
- ❁ Prerequisite for Database Design – Normalization
- ❁ Conventional File Design
- ❁ Modern Database Design
- ❁ The Next Generation of Database Design

Chapter Map



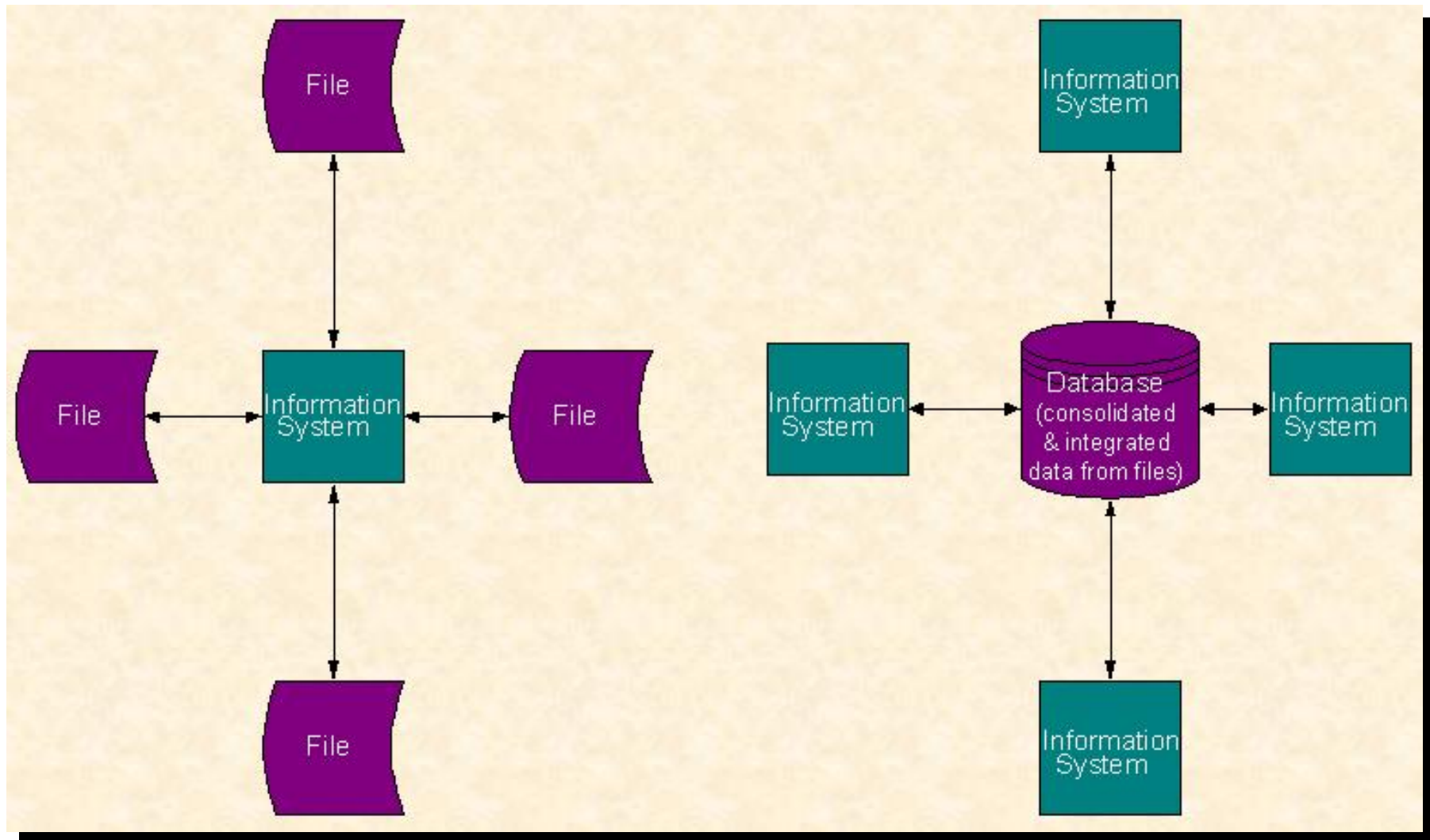


Conventional Files versus the Database

Files versus Database

- ❁ A **file** is a collection of similar records.
- ❁ A **database** is a collection of interrelated files (meaning that records in one file are physically related to records in another file).

Files versus Database



Pros and Cons of Conventional Files

Pros

- ✿ Easy to design because of their single-application focus.
- ✿ Excellent performance due to optimized organization for a single application.

Cons

- ✿ Harder to adapt to sharing across applications.
- ✿ Harder to adapt to new requirements.
- ✿ Need to duplicate attributes in several files.

Pros and Cons of Databases

Pros

- ✿ Ability to share data across applications.
- ✿ Less, and controlled redundancy (total non-redundancy is not achievable).
- ✿ Data independence from applications increases adaptability.
- ✿ Superior scalability

Cons

- ✿ More complex than file technology.
- ✿ Somewhat slower performance.
- ✿ Investment in DBMS and database experts.
- ✿ Greater data vulnerability



Database Concepts for the Systems Analyst

Fields

- ✿ A **field** is the physical implementation of a data attribute. They are the smallest unit of meaningful data.
 - A **primary key** is a field whose values identify one and only one record in a file.
 - A **secondary key** is an alternate identifier for a record.
 - A **foreign key** is a pointer to a record in a different file.
 - A **descriptive field** is any other (nonkey) field that stores business data.

Records

- ✿ A record is a collection of fields arranged in a predefined format.
 - Fixed-length record structures
 - Variable-length record structures
- ✿ A **blocking factor** (组块因数) is the number of logical records included in a single read or write operation (from the computer's perspective). A block is sometimes called a physical record.

Files

- ✿ A **file** is the set of all occurrences of a given record structure.
- ✿ A **table** is the relational database equivalent of a file.
 - File and table types
 - Master files – contain records that are relatively permanent.
 - Transaction files – contain records that describe business events.
 - Document files – contain stored copies of historical data.
 - Archival files – contain master and transaction file records that have been deleted from on-line storage.
 - Table lookup files – contain relatively static data that can be shared by applications.
 - Audit (审计) files – are special records of updates to other files. They are used in conjunction with archival files to recover “lost” data.
 - File organization – how the records should be sequenced
 - File access – how the records should be accessed

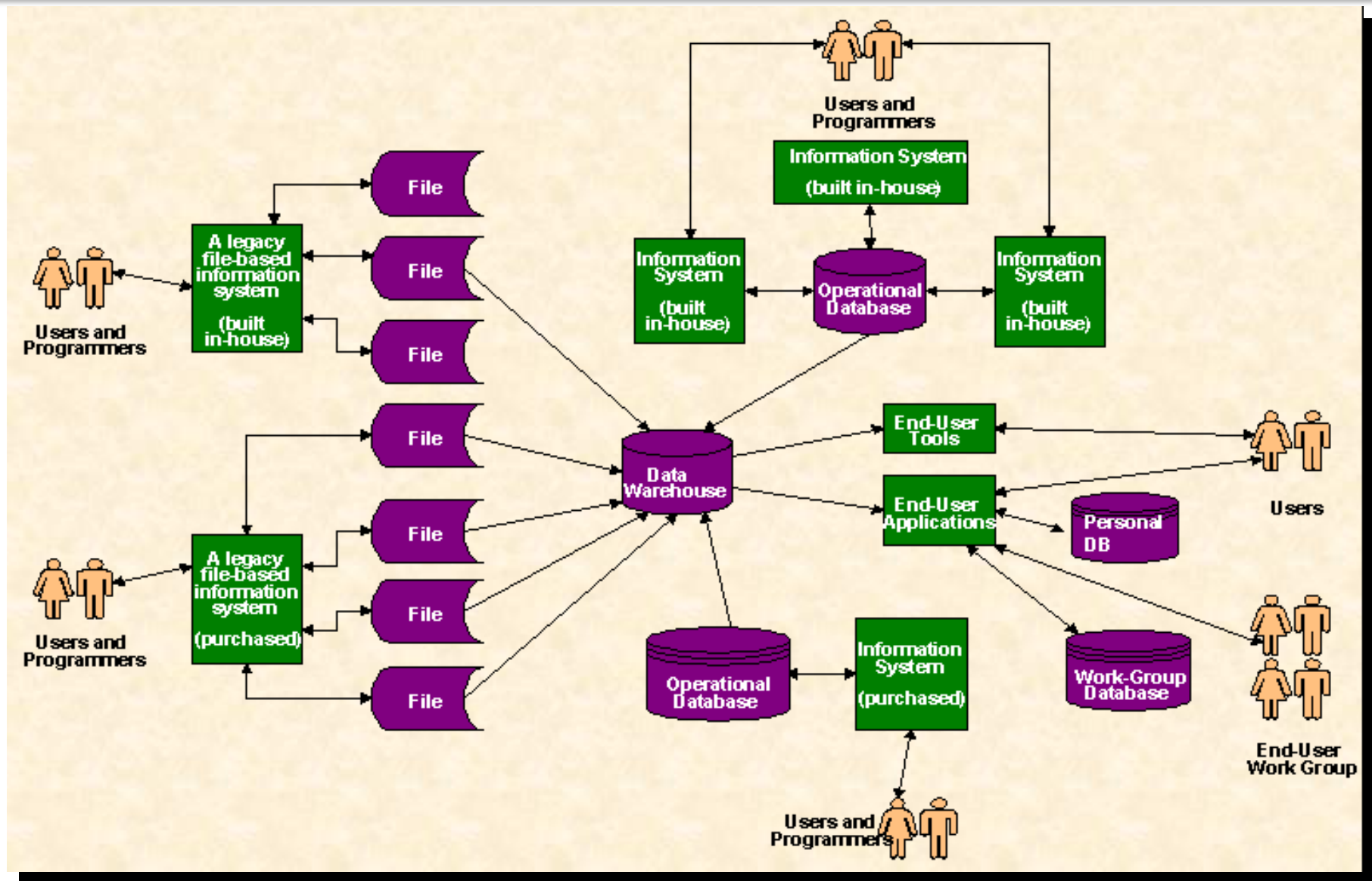
Data Architecture

❁ A business's data **architecture** defines how that business will develop and use files and databases to store all of the organization's data; the file and database technology to be used; and the administrative structure set up to manage the data resource.

❁ Data is stored in some combination of:

- **Conventional files**
- **Operational databases** (also called transactional databases)
- **Data warehouses**
 - To support data mining
- **Personal databases**
- **Work group databases**

A Modern Data Architecture



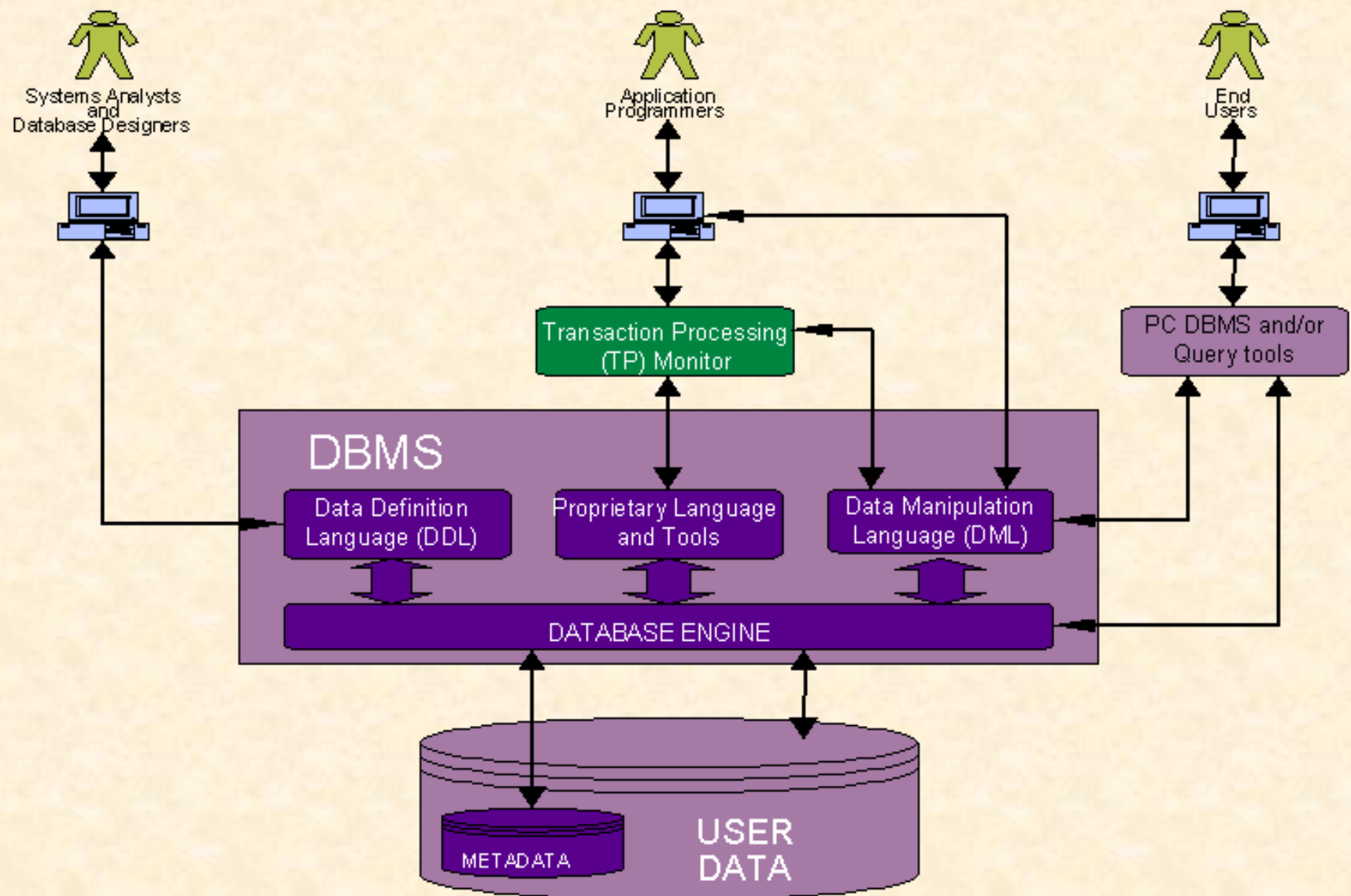
Administrators

- ❁ A **data administrator** is responsible for the data planning, definition, architecture, and management.
- ❁ One or more **database administrators** are responsible for the database technology, database design and construction consultation, security, backup and recovery, and performance tuning.

Database Architecture

- ❁ **Database architecture** refers to the database technology including the database engine, database utilities, CASE tools, and database development tools.
- ❁ A **database management system** (DBMS) is specialized software that is used to create, access, control, and manage the database. The core of the DBMS is a database engine.
 - A **data definition language** (DDL) is used to physically define tables, fields, and structural relationships.
 - A **data manipulation language** (DML) is used to create, read, update, and delete records in the database, and navigate between different files (tables) in the database.

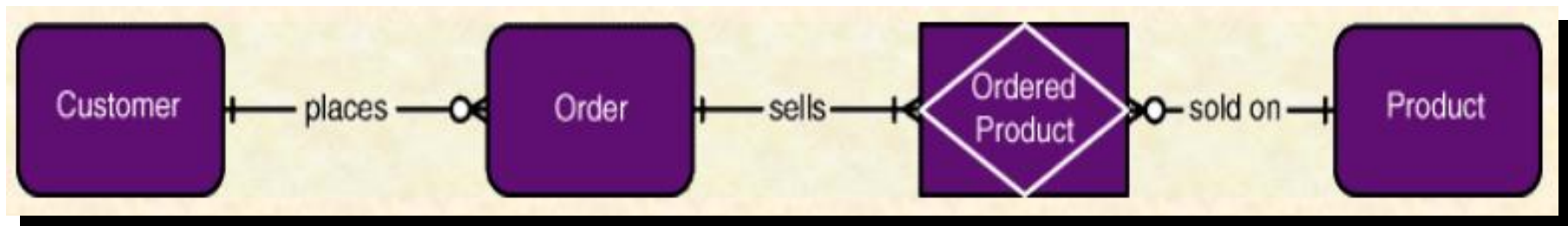
Typical DBMS Architecture



Relational Databases

- ✿ Relational databases implement stored data in a series of two-dimensional tables that are “related” to one another via foreign keys.
 - The physical data model is called a **schema**.
 - The DDL and DML for a relational database is called **SQL** (Structured Query Language).
 - **Triggers** are programs embedded within a table that are automatically invoked by updates to another table.
 - **Stored procedures** are programs embedded within a table that can be called from an application program.

Typical DBMS Architecture



Physical Data Model (Relational Schema)

Customers Table

Customer Number (primary key)	Customer Name	Customer Balance	...
10112	Luck Star	1455.77	
10113	Pemrose	12.14	
10114	Hartman	0.00	
10117	K-Jack Industries	- 20.00	

Orders Table

Order Number (primary key)	Customer Number (foreign key)	...
A633	10112	
A634	10114	
A635	10112	

Ordered Products Table

Order Number (foreign key)	Product Number (foreign key)	Quantity Ordered	...
A633	77F02	1	
A633	77B12	500	
A634	77B13	100	
A634	77F01	5	
A635	77B12	300	
A635	77B15	15	

Products Table

Product Number (primary key)	Product Description	Quantity in Stock	...
77B12	Widget	8000	
77B13	Widget	0	
77B15	Widget	52	
77F01	Gadget	20	
77F02	Gadget	2	



Prerequisite for Database Design – Normalization

Data Normalization (also see Chapter 7)

- ❁ An logical entity (or physical table) is in **first normal form** if there are no attributes (fields) that can have more than one value for a single instance (record).
- ❁ An logical entity (or physical table) is in **second normal form** if it is already in first normal form and if the values of all nonprimary key attributes are dependent on the full primary key.
- ❁ An logical entity (or physical table) is in **third normal form** if it is already in second normal form and if the values of all nonprimary key attributes are not dependent on other nonprimary key attributes .



Modern Database Design

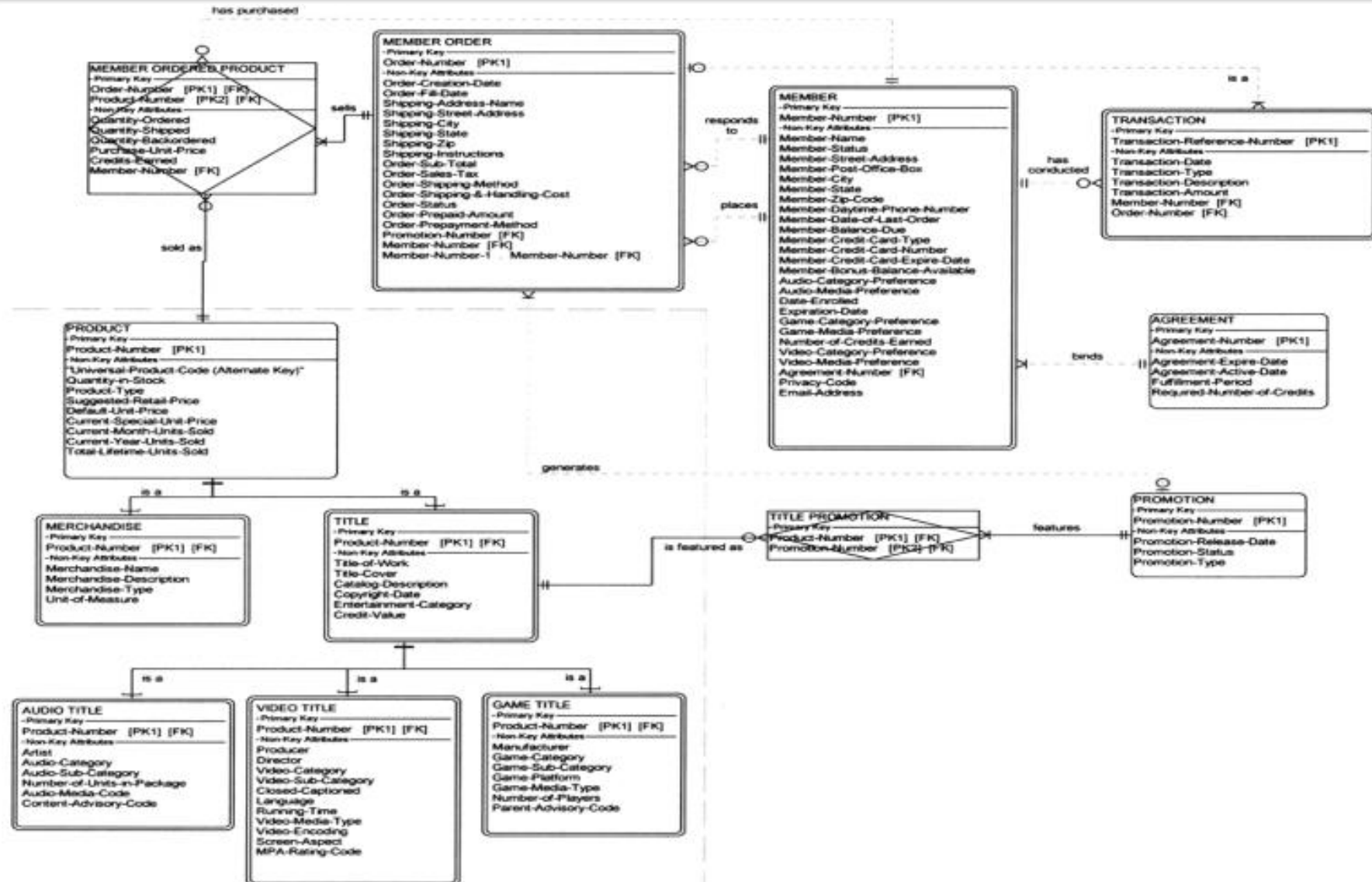
Goals of Database Design

- ❁ A database should provide for efficient storage, update, and retrieval of data.
- ❁ A database should be reliable – the stored data should have high integrity and promote user trust in that data.
- ❁ A database should be adaptable and scalable to new and unforeseen requirements and applications.

Data and Database Models

- ✿ An **entity relationship diagram** is the logical model of the data requirements.
 - See chapter 7.
- ✿ A **database schema** is the physical model or blueprint of the planned implementation of the logical model.
 - Also called a **physical data model**.

SoundStage Logical Data Model



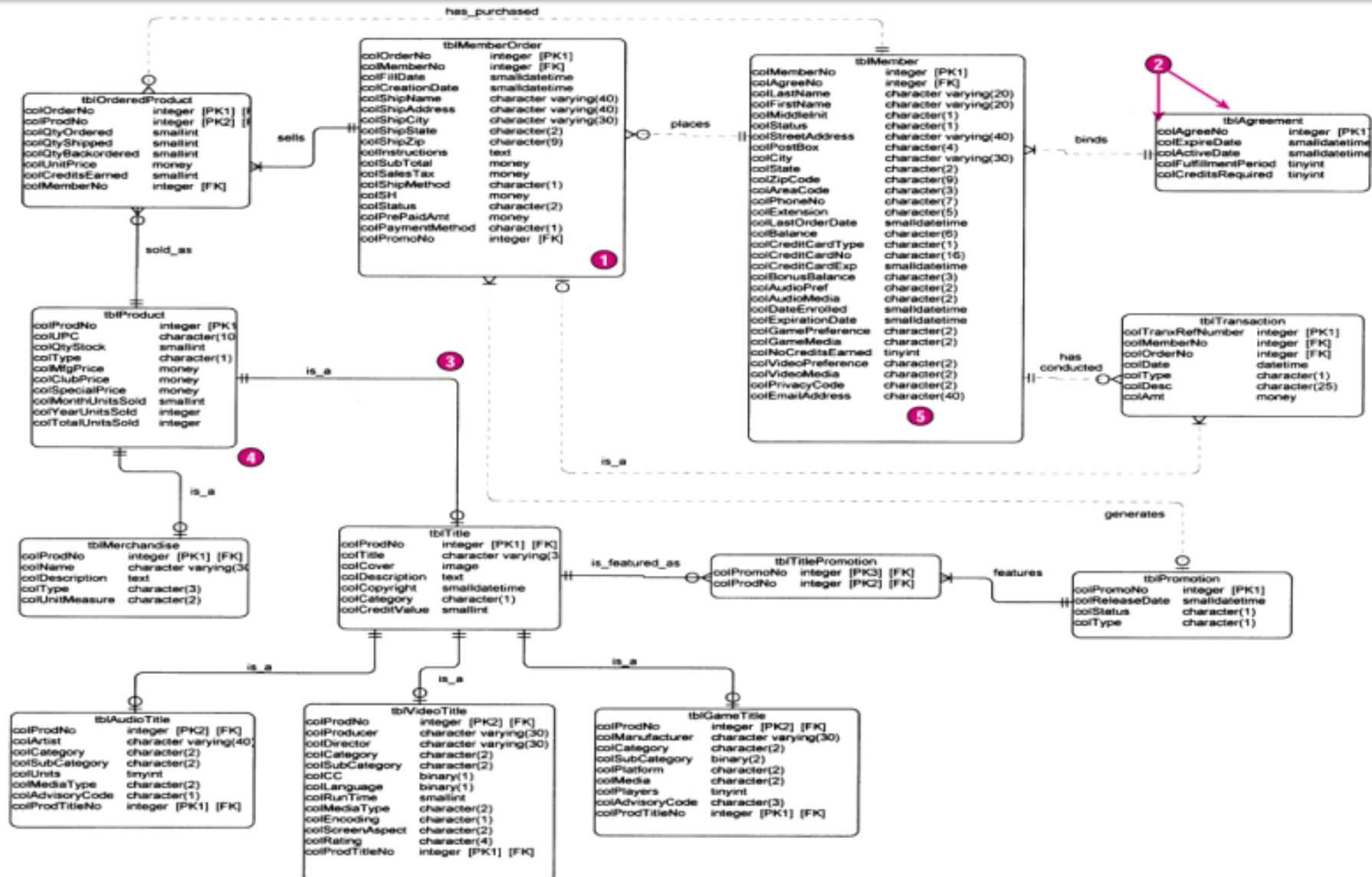
Sample Physical Data Types

Logical Data Type to be stored in field)	Physical Data Type Microsoft Access	Physical Data Type Microsoft SQL Server	Physical Data Type Oracle
Fixed length character data <i>(use for fields with relatively fixed length character data)</i>	TEXT	CHAR (size) or character (size)	CHAR (size)
Variable length character data <i>(use for fields that require character data but for which size varies greatly--such as ADDRESS)</i>	TEXT	VARCHAR (max size) or character varying (max size)	VARCHAR (max size)
Very long character data <i>(use for long descriptions and notes--usually no more than one such field per record)</i>	MEMO	TEXT	LONG VARCHAR or LONG VARCHAR2
Integer number	NUMBER	INT (size) or integer or smallinteger or tinyinteger	INTEGER (size) or NUMBER (size)
Decimal number	NUMER	DECIMAL (size, decimal places) or NUMERIC (size, decimal places)	DECIMAL (size, decimal places) or NUMERIC (size, decimal places) or NUMBER

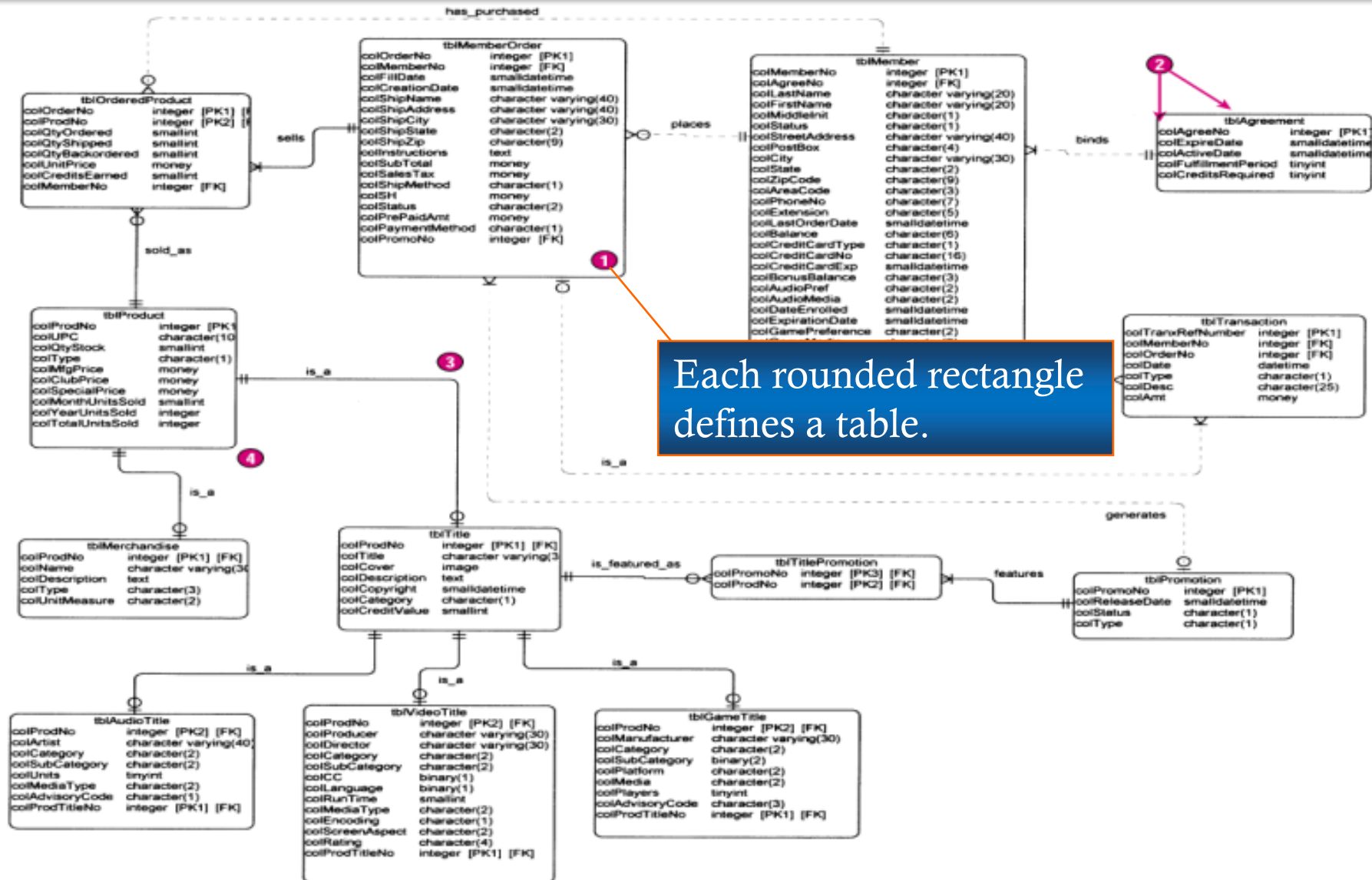
Sample Physical Data Types (concluded)

Logical Data Type to be stored in field)	Physical Data Type Microsoft Access	Physical Data Type Microsoft SQL Server	Physical Data Type Oracle
Financial Number	CURRENCY	MONEY	<i>see decimal number</i>
Date (with time)	DATE/TIME	DATETIME <i>or</i> SMALLDATETIME <i>Depending on precision needed</i>	DATE
Current time (<i>use to store the data and time from the computer's system clock</i>)	<i>not supported</i>	TIMESTAMP	<i>not supported</i>
Yes or No; or True or False	YES/NO	BIT	<i>use CHAR(1) and set a yes or no domain</i>
Image	OLE OBJECT	IMAGE	LONGRAW
Hyperlink	HYPERLINK	VARBINARY	RAW
Can designer define new data types?	NO	YES	YES

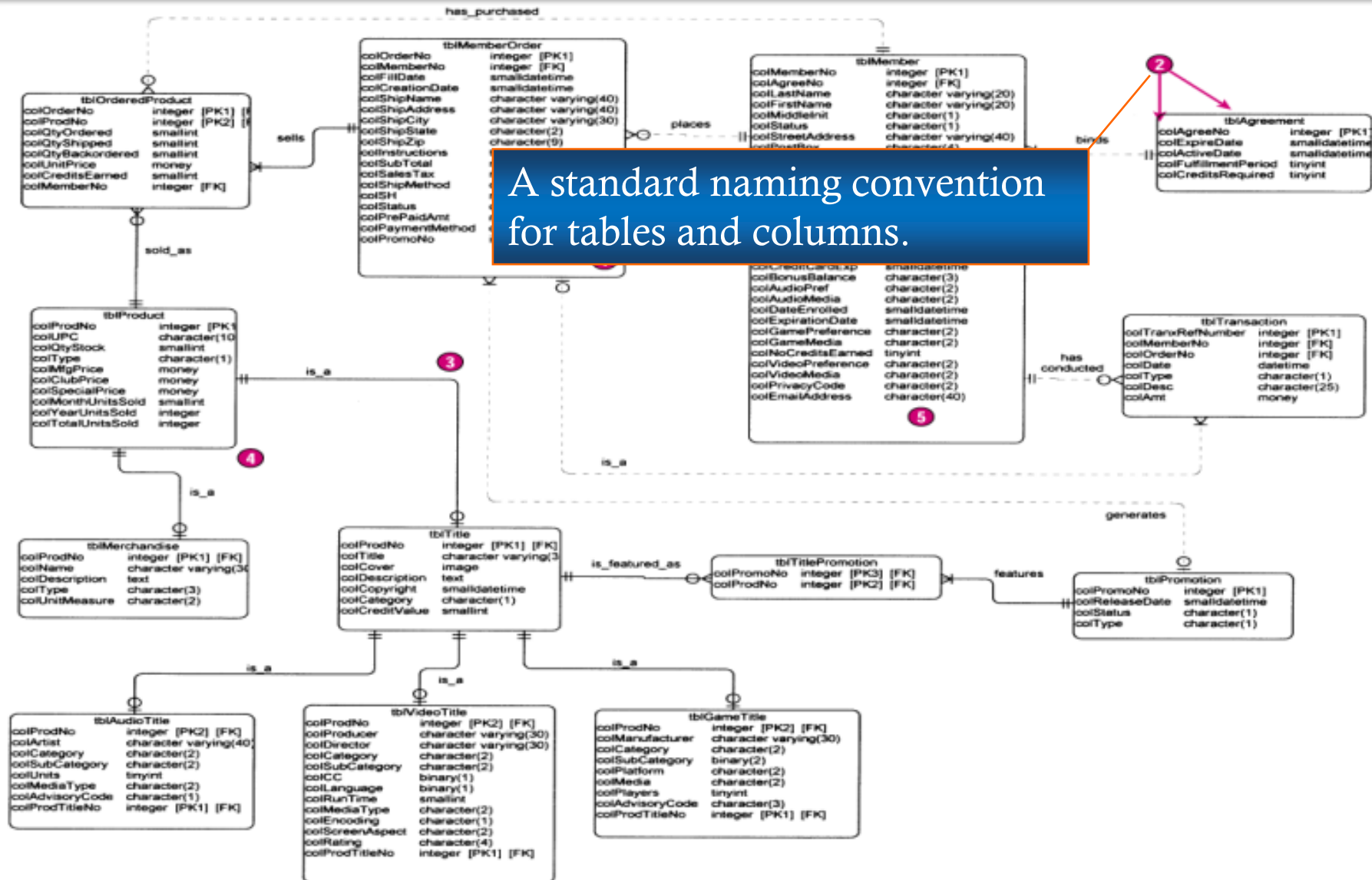
SoundStage Physical Database Schema



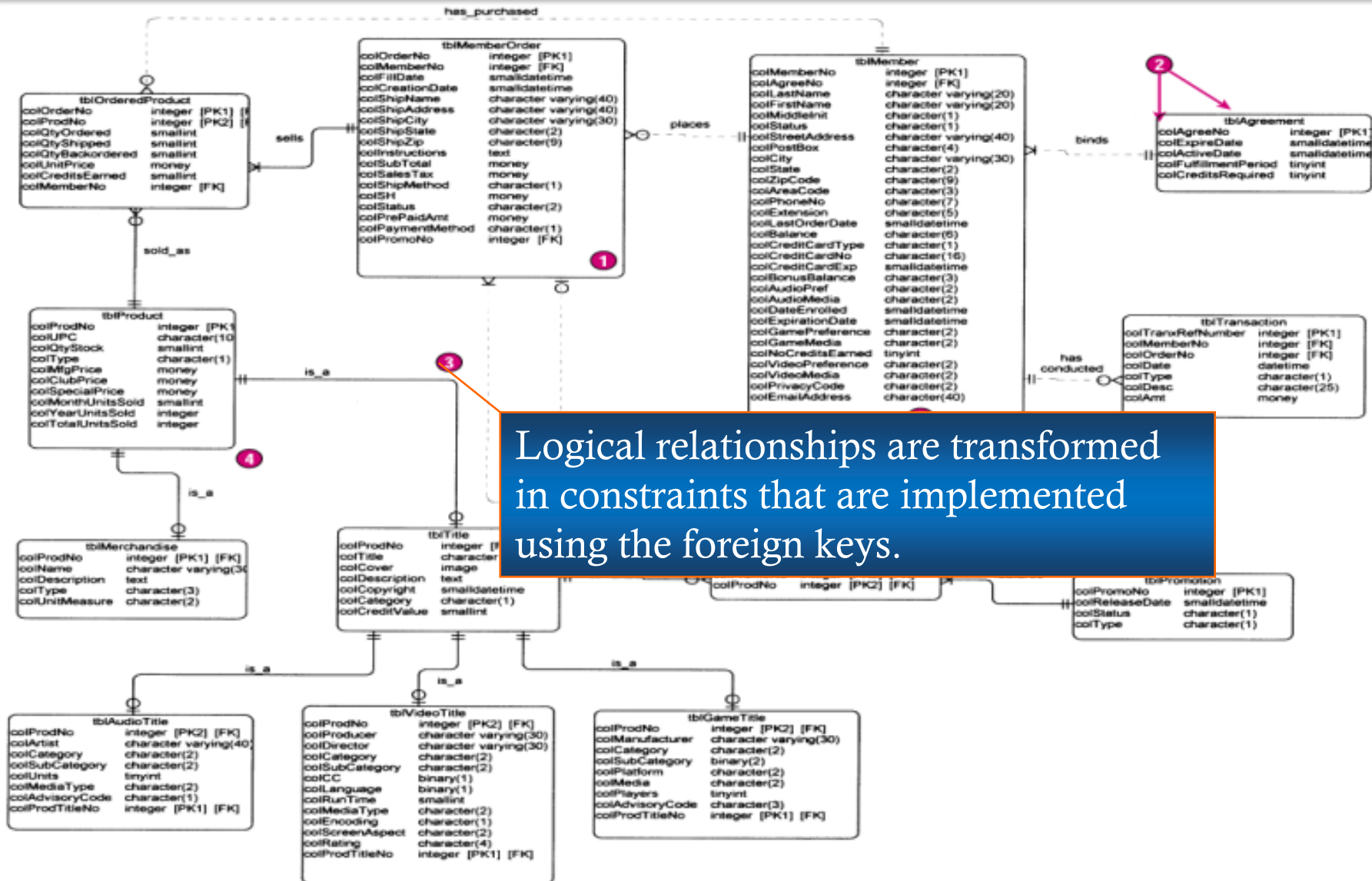
SoundStage Physical Database Schema



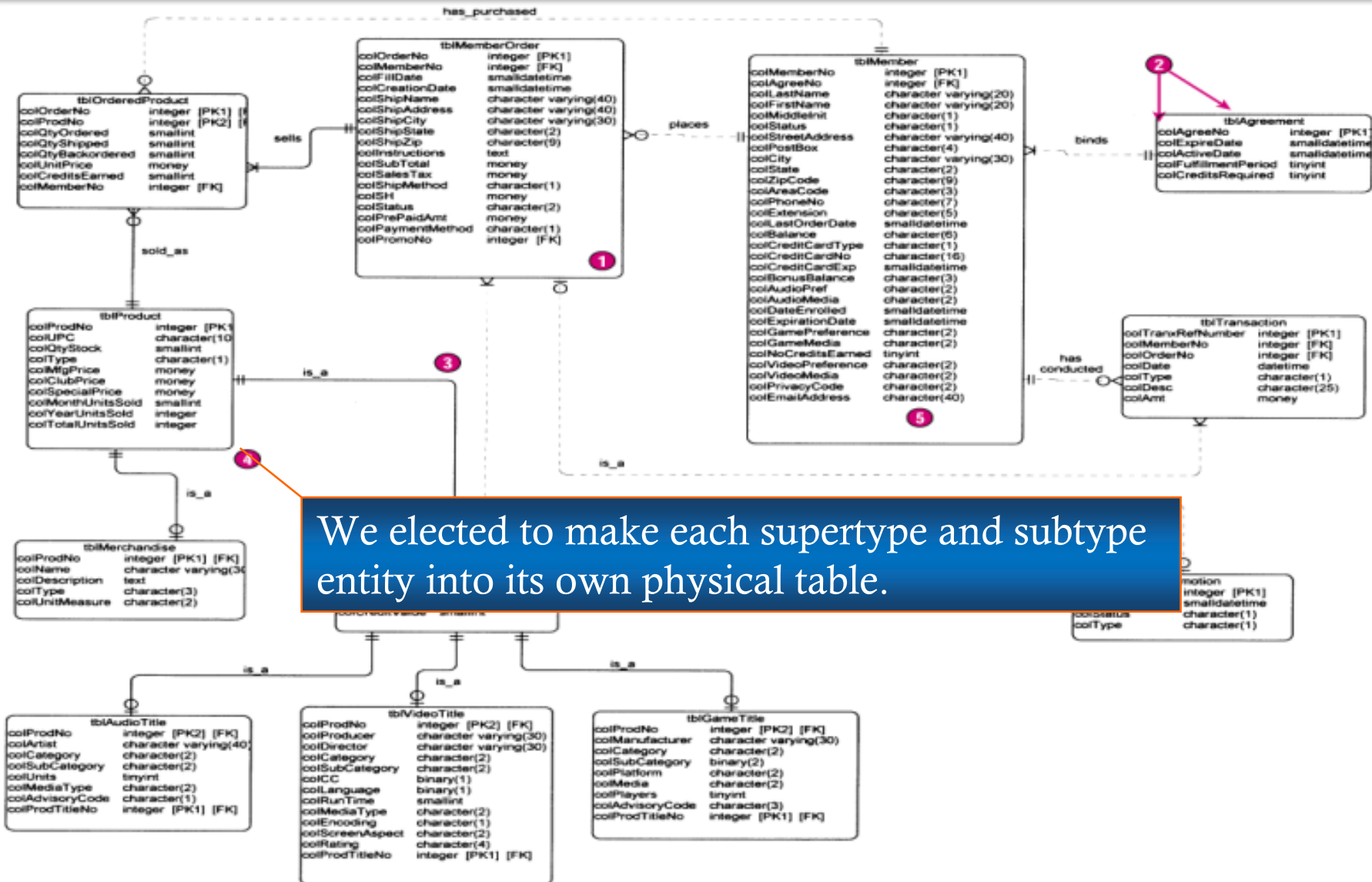
SoundStage Physical Database Schema



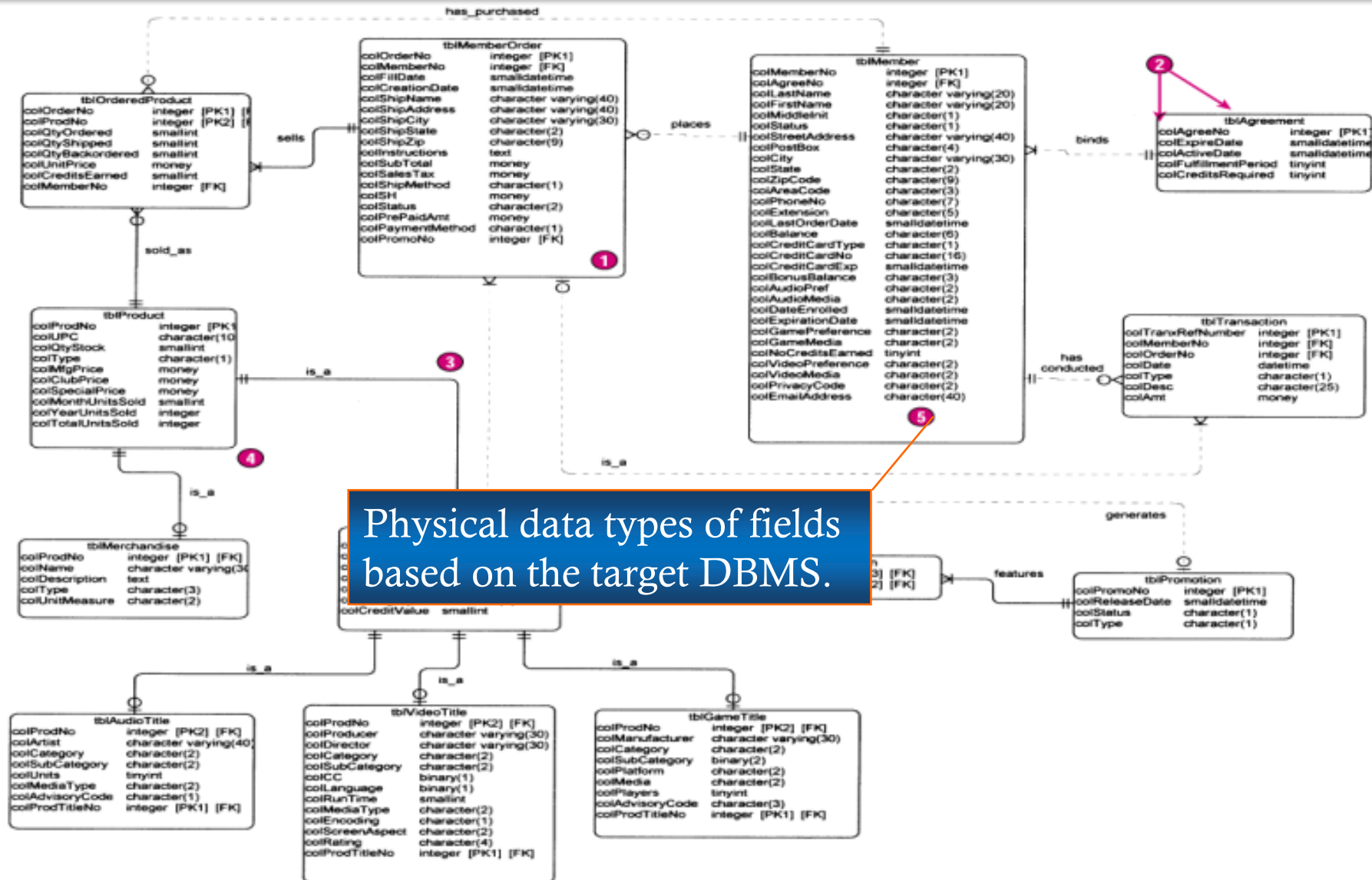
SoundStage Physical Database Schema



SoundStage Physical Database Schema



SoundStage Physical Database Schema



A Method for Database Design

- ✿ Review the logical data model.
- ✿ Create a table for each entity.
- ✿ Create fields for each attribute.
- ✿ Create an index for each primary and secondary key.
- ✿ Create an index for each subsetting criterion.
- ✿ Designate foreign keys for relationships.
- ✿ Define data types, sizes, null settings, domains, and defaults for each attribute.
- ✿ Create or combine tables to implement supertype/ subtype structures.
- ✿ Evaluate and specify referential integrity constraints.

Database Integrity

❁ Key integrity (关键字完整性)

- Every table should have a primary key.
- The primary key must never be allowed to have a null value.

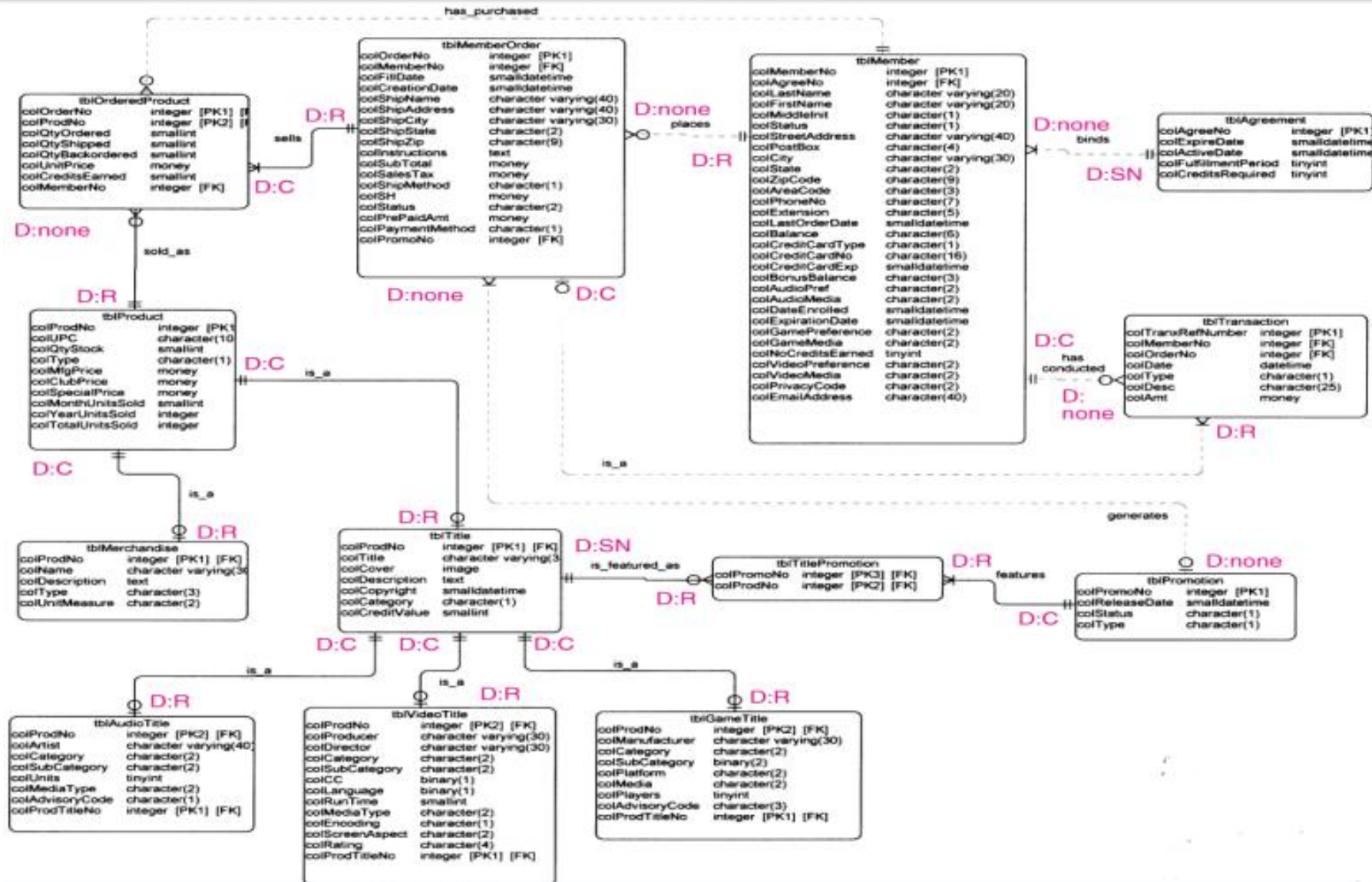
❁ Domain integrity (定义域完整性)

- No field takes on a value that is outside of the range of legal values.

❁ Referential integrity (参照完整性)

- A **referential integrity** error exists when a foreign key value in one table has no matching primary key value in the related table.
- Deletion rules:
 - No restriction
 - Delete: Cascade (进行梯次删除)
 - Delete: Restrict (从相关的表中删除, 才能删除该记录)
 - Delete: Set null (将相关记录中的域置为空值)
 - Delete: Set Default (将相关记录中的域置为缺省值)

SoundStage Physical Database Schema



Database Distribution and Replication

✿ **Data distribution analysis** establishes which business locations need access to which logical data entities and attributes.

- The analysis drives distribution decisions:
 - Centralization
 - Horizontal distribution (also called partitioning)
 - Vertical distribution (also called partitioning)
 - Replication

要点与引申

- ❁ 本章中所提到的文件，实际上都是指记录文件。
- ❁ 命名标准化是一个企业的软件开发规范程度的体现。
- ❁ 为提高性能而采用必要的冗余与保持数据一致性，始终是关系数据库设计的关键问题之一。