INFS3110 – Accounting Information Systems

Week 1: Databases (FA)

Accounting Database collects data about: people, things, transactions.

Transactional Info: Records every business event accurately, maintains a complete audit trail and ensures the historical financial record is reliable.

- Supports financial accounting and includes airline tickets, sales receipts and packing slips.
- This data directly underpins the preparation of financial statements, the SOPL, SOFP, and SOCF, and supports external reporting obligations defined by law and regulation.
- Completeness and accuracy of transactional data is therefore critical to meeting f/s assertions such as occurrence and completeness.

Analytical Info: Provides processed insights needed for internal decision-making, budgets, price and eff variances, CBA and performance dashboards.

- Includes product statistics, sales projects, future growth and trends. It supports management accounting.
- Defined by management needs rather than regulatory requirements.
- A robust accounting database must integrate both types so orgs can maintain reliable financial records while also using that same underlying data to drive better internal decisions.

User & Application Tiers

1. **Database Tier**: Contains Operational Database and DBMS. **Stores, manages and structures raw data.** Focus on structure of operational database (commonly relational databases), query language and sometimes data warehouses in business intelligence.
2. **Application Tier**: Shows accounting software (e.g. SAP). Acts as the middle layer that connects the database to the user interface. Processes data from the database and provides functionality tailored to accounting tasks. **Processes and interprets data.**
3. **User Tier**: Depicts the acctg interface (so forms and reports on computer screens). Layer where end-users interact with the system. Users don't see database directly; they work with reports, dashboards and forms generated by the application tier. **Presents data in usable formats.**

Advantages of Database Systems:

+ **Eliminates data repetition and inconsistency:** Relational database stores each piece of data in one place, ensuring any update is immediately reflected everywhere it is referenced. Prevents errors in journals, GL, f/s.
+ **Increased flexibility and speed**: Can query large volumes of transaction data across multiple dimensions (by period, dept, supplier or product) in seconds). Supports timely management reports.
+ **Improves system security:** Enabling RBAC, automated audit logs and enforced approval workflows. Database system ensures only authorised users can access sensitive acctg data, supporting control environment and compliance requirements (e.g. Sarbanes Oxley).

Data Models: Used to describe and represent complex real-world data structures. ERM is most common.

- Involves identifying entities, capturing attributes, defining relationships and reflecting business rules.
- Create a structured plan that guides the physical implementation of the database by organising data into logical structures such as tables and keys, and ensuring data and referential integrity.
- Also supports efficient data storage and reduces data redundancies.

Symbols for Database Design (Chen/Crows Foot Notation)

- Rectangles → Entities
  - A thing of interest to the organisation. About which data may be stored. Can be a person, an object, an event, a concept.
- Ovals → Attributes
- Diamonds → Relationships

Entity Types & Instances

- ENTITY describes a named grouping of all entities that are the same.
  - Ex: All the people who enrol at uni belong to the entity: STUDENT.
- INSTANCE is one member of the named entity.
  - Ex: An instance of the entity STUDENT might be Mary Zhang.

Entity Attributes

- An identified characteristic (property) of an entity type.
- Attributes of STUDENT might include: Student_Name, Student_ID (PK), Birth_Date, Gender.
- KEY uniquely identifies an instance of an entity type

- KEY ATTRIBUTES are identified in an ERD by underlining the name of the attribute.
    - E.g. Primary Key for STUDENT could be Student_ID.
    - Some key attributes are made up of two or more fields in order to get a unique value, called a composite key.
    - A key attribute value should be fixed for the life of the entity.
    - A composite key is needed when no single column in a table can uniquely identify a record, but a combination of two or more columns together can. In a many-to-many relationship. Student_ID and Course_ID can uniquely identify enrollment but neither Student_ID or Course_ID can.

Relationships: An association among two or more of the entities. Represented by a line connecting two entities. Often identified with verbs/verb phrases.
Cardinality Relationships: Max no of times an instance of one entity can relate to instances of another entity. Examples: 1-1, 1-M, M-M relationship.

ONE-TO-MANY RELATIONSHIP

- Left to right: A degree is assigned to many students.
- Right to left: A student is assigned to one degree.

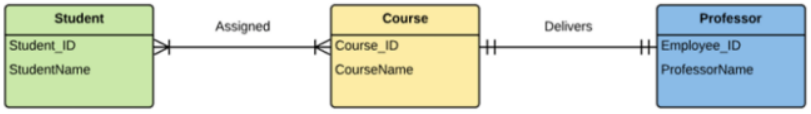
ADDING ORDINALITY (OPTIONAL)

- Ordinality: Minimum number of times an instance in one entity can be associated with an instance in the related entity.
- It shows whether the relationship is mandatory (short vertical line) or optional (circle).
- In the above diagram, a student MUST be assigned one or many courses, but a course MAY be assigned to students.

MANY-TO-MANY RELATIONSHIPS

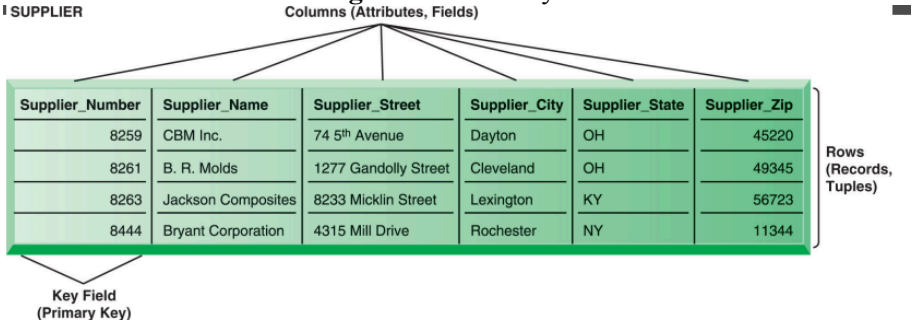
- Left to Right: A student is assigned to many courses.
- Right to Left: A course is assigned to many students.
- NOTE: Crow's Foot Notation means there could be one or many in the relationship.

Relationship Expanded



- Left to Right: A student is assigned to many courses. The course is delivered by one professor.
- Right to Left: A professor delivers one course that's been assigned to many students.
- NOTE: Crow's Foot (double line) means there's only one relationship possible here between the instances.

ER Model Maps → Relational Database Design → Each entity becomes a table. Each attribute becomes a field.



PART

Part_Number	Part_Name	Unit_Price	Supplier_Number
137	Door latch	22.00	8259
145	Side mirror	12.00	8444
150	Door molding	6.00	8263
152	Door lock	31.00	8259
155	Compressor	54.00	8261
178	Door handle	10.00	8259

Primary Key Forei n Key

Week 2: Database Relationships (FA II)

Foreign Key: A column in a database table that is linked to a column in a different table.

- Defines relationships between tables (hence relational database).
- Rows can only be added or updated in the child table if the value being updated matches a value in the 'parent' table. Builds relationships and sustains data integrity between two tables.

Why Relate Tables to One Another Again? Data stored in one large table can be redundant and inefficient causing the following problems:

1. **Update** anomaly: The same data is stored in multiple rows, so updating it requires changing every instance, miss one and you have inconsistent data.
2. **Insert** anomaly: You can't add new data without also supplying unrelated data (e.g. can't add a new supplier unless you also have an order to attach it to).
3. **Delete** anomaly: Deleting one record unintentionally destroys other useful data (e.g. deleting the only order for a supplier also wipes the supplier's details entirely).

Non-Normalised Data (only table the store uses)

Order#	Order Date	Customer#	Customer Name	Item#	Item Quantity	Item Price
123	11/12/07	101	Smith's Hi Fi	12	19	23.78
123	11/12/07	101	Smith's Hi Fi	14	2	145.99
123	11/12/07	101	Smith's Hi Fi	15	5	200.00
123	11/12/07	101	Smith's Hi Fi	23	1	96.00
126	11/12/07	102	Hi Fi Limited	11	2	67.89
126	11/12/07	102	Hi Fi Limited	14	3	145.99
127	12/12/07	101	Smith's Hi Fi	13	1	99.99
127	12/12/07	101	Smith's Hi Fi	14	2	145.99

Normalisation: A set of rules and a process of assigning attributes to entities to eliminate repeating groups and data redundancies, and to form tables representing entities that promote structural and data independence.

- **Top-down versus bottom-up design:**
  - ERD is a strategic top down way of developing a database model.
  - Normalisation is the bottom up view of designing a database (from scratch).
  - Once the table structure is defined, normalisation is used to assign attributes to entities.
  - Both techniques are used in the design process.

Database Tables Reduce These Instances

- **Data Redundancy**: Exactly the same data is recorded and stored in different location which can lead to data inconsistency and anomalies.
- **Data Integrity:** Data that provides a consistent and correct representation regardless of where it is sourced from within a file system.

Why Query the Database? To retrieve stored data. A query is a structured request for info from the database. We can make good decisions from stored data and info provided by the query.

DBMS & Accounting

- Databases expand accounting's ability to produce real-time dynamic reports.
- DBMS's are capable of integrating financial, operational and external data.
  - E.g. Customer satisfaction data can be stored for a richer set of data for decision-making.
- Understanding database systems is an enabler for accountants.
- Participation important for ensuring adequate controls are in place to safeguard data and ensure reliability.

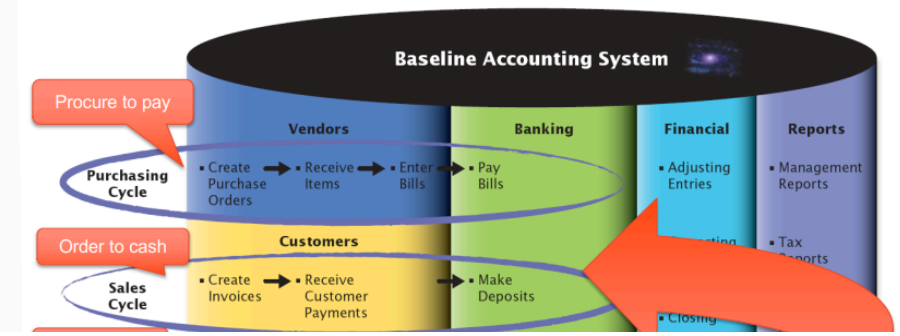
Week 3: Business Processes (FA III)

Business Process: A collection of linked tasks that find their end in the delivery of a service or product to a client.

- Info systems (support for bz process) and, therefore databases are built on business processes.
- Activities that work together to achieve bz objectives.
- The process must involve…inputs and a single output. These inputs are made up of all the factors that contribute…to the added value of a service or product.

Transaction Cycles of the Baseline Accounting System (Sales & Purchasing Examinable)

+ **Sales -** Order to Cash, Order to Sales
+ **Purchasing** - Procure to Pay, Purchasing Transactions
+ **Payroll** - Employee Transactions



Key Takeaways

- BPs inevitably involve financial touchpoints - master data, event data, in the accounting system.
- Financial implications means accountants need to be involved.
- The drawings - different types of drawings perhaps not the most efficient way to do this.
- Does each step represent value to one of the stakeholders?
- It's the financial data that is impacted. So how do we manage the risks?

System Documentation: When a BP is documented the interactions with the IS enabling that process are shown.

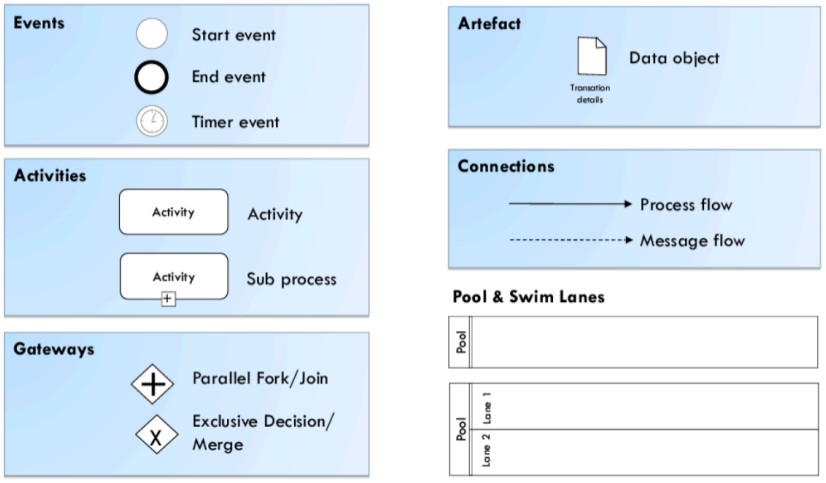
- Way of visually depicting the operations of a system. Important throughout acctg career.

Purpose of Systems Documentation: Depicting how the system works, training users, designing new systems, auditing AIS, compliance, documenting BPs, streamlining tool, standardising communication w others.

Key piece of knowledge that an accountant can offer the organisation in situations of: Implementing and designing new systems, or adopting system modifications, or the need for well designed internal controls.

During systems development the accountant (and the auditor) should review documentation to preserve the knowledge of a process. System documentation is important in preserving the corporate memory of the process.

BPMN Process Model: Diagram consisting of four key types of elements.



Advantages & Disadvantages of BPMN

+ **Supports process automation**: BPMN models can be directly translated into executable workflows in BPMS. This makes it suitable not only for documentation but also for implementation.
+ **Enhances process analysis and improvement:** By visualising the entire process, organisations can identify bottlenecks, redundancies and inefficiencies. Supports continuous improvement and lean initiative.
+ **Widely supported by tools:** Many modelling platforms (e.g. Signavio, Visio) support BPMN, making it easy to create, share and maintain diagrams.
- **Can be overly complex:** When modellers try to capture every detail, diagrams can be difficult to interpret. This defeats the purpose of clarity and may confuse stakeholders.
- **Requires stakeholder discipline:** For BPMN to be effective, all teams must follow the notation consistently. In organisations with low process maturity, this can be difficult to enforce.

Week 4: Application Controls (FA IV)

Application Controls: May be application/hybrid/manual controls.

Objectives of Application Controls

+ Input data is accurate, complete, authorised and correct.
+ Data is processed as intended in an acceptable time period.
+ Data stored is accurate and complete.
+ Outputs are accurate and complete.
+ A record is maintained to track the process of data from input to storage and to the eventual output.

Core Principles of Application Controls: Accuracy, completeness, validity, authorization, classification, timeliness, processing integrity (ensures data is processed consistently and completely from input to output).

Types of Application Controls:

1. **Input** controls: Prevent incorrect or incomplete data from entering the system (e.g. edit checks, drop-down menus and mandatory fields).
2. **Processing** controls: Ensure data is processed accurately and completely (e.g. batch totals, cross-footing, automated exception reports).
3. **Output** controls: Ensure reports and results are accurate, complete and distributed securely (e.g. distribution lists, report validation, audit trails).
4. **Master data** controls: Protect the accuracy and integrity of key reference data (e.g. restricted access, change logs, dual approval for updates).

Sales Cycle: BP by which a company receives and fulfils a customer order and collects payment.

- Flow: Customer places order → company processes and approves order → goods are picked, packed and shipped → invoice is sent to customer → customer pays → payment is matched and posted to accounts.
- Key Participants: Sales Rep, Warehouse Supervisor, Warehouse Clerk, A/R Clerk and Cost Accountant.

Control Objectives – Sales Cycle

1. **Process Customer Orders**
   - Process orders only within approved customer credit limits.
   - Process customer orders accurately.
   - Process only valid customer orders.
   - Transfer order data completely and accurately to shipping and accounts receivable.
2. **Process Shipments**
   - All shipments are accurately documented and data transferred to accounts receivable.
   - Ship customer orders in a timely manner.
3. **Process Accounts Receivable**
   - All goods shipped are accurately billed in the proper period.
   - Accurately record invoices for all authorised shipments.
   - Ensure completeness and accuracy of accounts receivable.
   - Safeguard accounts receivable records.
4. **Process Funds (Record Customer Payments)**
   - Record cash receipts on accounts receivable completely and accurately.
   - Monitor timely collection of accounts receivable.

Example of Related Control Activities: SOD → person responsible for custody of cash received and bank deposits is not the same person who is responsible for recording customer cash received in the acctg records. Any possible deposits withheld would then be detected unless the acctg records were falsified.

Purchasing Cycle (AKA Procure-to-Pay): BP by which an organisation acquires g/s from vendors and settles the resulting financial obligations. It's a feeder process into the General Ledger and directly impacts the accounts payable balance, inventory and cash disbursements on the financial statements.

Key Participants

Role	Responsibility
Requisitioner/Department	Identifies the need; raises a purchase requisition
Purchasing Officer	Reviews requisition, selects vendor, creates Purchase Order (PO)
Approver/Manager	Authorises the PO before it is sent to vendor
Warehouse/Receiving Dept	Receives goods; creates Goods Receipt (GR)
Accounts Payable Clerk	Matches invoice to PO and GR; enters vendor invoice
Finance/Treasury	Manages payment timing and cash disbursements
Vendor/Supplier	Provides g/s; sends invoice

End-to-End Process Flow (Sequence)

- Purchase Requisition raised by department → approved internally
- Purchase Order (PO) created by Purchasing and sent to vendor – this is a legal commitment
- Vendor delivers g/s → receiving department creates a Goods Receipt (GR)
- Vendor sends invoice → accounts payable receives and enters it
- Three-way match performed: PO vs GR vs Invoice must all agree on quantity, price and vendor.
- Invoice posted to accounts payable; payment scheduled
- Payment disbursed to vendor; GL updated.

Three-way match is the key control in the purchasing cycle. If all three documents agree, payment is authorised. Discrepancies are held for investigation.

Control Objectives – Purchasing Cycle

1. **Process Purchases**
   - Order items meeting specifications – right quality, right vendor
   - Receive items ordered on a timely basis
   - Record authorised purchase orders completely and accurately
   - Prevent unauthorised use of purchase orders – e.g. consecutive pre-numbering of POs
2. **Process Accounts Payable**
   - Accurately record bills on a timely basis for all authorised and accepted purchases.
   - Accurately record returns and allowances for all authorised credits.
   - Ensure completeness and accuracy of accounts payable.
   - Safeguard accounts payable records.
3. **Pay Bills**
   - Manage timing of cash disbursements.
   - Disburse cash only for authorised purchases – e.g. matching PO, GR and invoice.
   - Record cash disbursements completely and accurately.

Application Controls: Automated controls built into the system (e.g. SAP) that enforces control objectives above.

Control Objective	Example: Application Control
Prevent unauthorised POs	System requires manager e-approval before PO is issued
Accurate recording of goods receipt	SAP GR transaction auto-updates inventory and triggers liability
Three-way match	System blocks invoice posting if PO/GR/invoice values don't reconcile
Segregation of duties (SOD)	Role permission table prevents same user creating PO and approving payment
Prevent duplicate payments	System checks invoice number against existing records
Cut-off accuracy	System date-stamps all entries; prevents backdating

Segregation of Duties (SOD): Key control. Four functions that must be separated are: Authorisation (approving transaction), recording (entering data into the acctg system), custody (physical control of assets) and reconciliation (comparing records to physical assets).

Relevance to Financial Accounting:

- Accounts Payable (liability): Created when invoice is posted.
- Inventory/Expense: Debit entry on goods receipt.
- Cash: Reduced when payment is made.
- The completeness assertion is directly supported by three-way match controls.
- The existence/occurrence assertion is supported by requiring a PO before any purchase.

Typically uses a role permission table: Offers a matrix style environment to manage system access permissions. It assigns roles.

Business processes will have an element of inherent risk regardless of the application used to support them, for example collusion between two staff members.

- Given tech is mainly used to support BPs, the label application control is used as an overarching term.
    - For example, transactional applications, such as SAP, support procure-to-pay processes and order-to-cash processes.
    - Risks may stem from:
        - How the system is configured, managed and used by employees to support these processes.
        - Inherent nature of the software application itself.

Accountant may be considered a:

- Designer of systems: Involving the collection, recording and storage of financial and acctg data.
- User of systems: Need for relevant and reliable info for financial reporting and beyond.
    - What are the financial statement assertions again?
        - E.g. Existence and occurrence – all reported transactions actually occurred during the reporting period and all assets and liabilities exist as of the reporting date.
- Auditor of systems: Internal control assessments.
    - Control design effectiveness – are the objectives being served.
    - Over financial reporting – e.g. Sarbanes Oxley requirements.

Risk Assessment: Aassessing the extent to which events would impact an entity's ability to achieve its objectives.

Two aspects to risk assessment:

- Impact: Effect that an event will have on the entity's ability to achieve its obj if the event occurs.
- Likelihood: Possibility or probability that a potential event will occur.

Week 5: Integrated Enterprise System & Silo Architecture (FA V)

Integrated Enterprise System:

- **Shared data structure**: An IES connects data across all functional areas (HR, Accounting, Sales, Ops, Supply Chain) via a single shared database.
- **Value chain integration**: Because modules like SCM, OPS, CRM, and Accounting are interwoven, they can exchange data electronically across the entire value chain.
- **Automated flow**: This electronic exchange means data moves automatically through automated workflows and business processes, no manual rekeying between departments.

Silo Architecture

- Refers to a system or organisational structure where processes, data, and technologies are isolated within individual departments or functional units (supply chain, production, HR, sales, accounting).
- Each "silo" operates with its own systems, databases, and workflows, with limited integration or communication across the organisation.
- This leads to fragmented information flows, duplicated effort, inconsistent data definitions, and reduced visibility across end-to-end processes.
- While silos can provide autonomy and specialised focus, they often hinder collaboration, slow decision-making, and create inefficiencies because information cannot move seamlessly between units.

Key Characteristics of Silo Architecture

- **Isolated systems** — each department uses its own applications, tools, and data stores.
- **Limited data sharing** — information is not easily accessible outside the silo.
- **Redundant processes** — similar tasks are repeated across units due to lack of integration.
- **Inconsistent data definitions** — each silo may maintain its own version of key data (e.g., customer, product).
- **Fragmented decision-making** — leaders lack a unified view of operations

Silo Architecture with Spaghetti Code

- Impossible integrations: Silos don't align, spaghetti code makes them unstable to connect.
- Data inconsistency: Each silo stores its own version of the truth.
- Operational inefficiency: Manual rekeying, reconciliation and duplicated effort.
- High maintenance cost: Every change requires testing across multiple fragile links.
- Shadow data growth: Employees create spreadsheets and personal databases to compensate.
- Blocked digital transformation: Modernisation becomes risky and expensive.

Silo Architecture with Spaghetti Code: Tiers