

플랫폼 개발 생명주기(3)

(PDLC: Platform Development Life Cycle)

3 단계 : 설계 (물리적 설계)

- ✓ 물리적 설계 = 사용자 인터페이스 + 데이터베이스 + 플랫폼 구조도
- ✓ 사용자 인터페이스(User Interface, UI)의 특징
 - UI는 사용자와 플랫폼 간 상호작용을 가능하게 해주는 도구
 - 일관성 있는 UI 디자인은 사용자의 사용률을 높이고 시스템을 이용하는 데에 있어서 혼란을 줄임
- ✓ UI 설계 시 고려사항
 - UI는 사용자가 플랫폼을 사용할 때 혼란을 겪지 않게 하기 위해서 일관성 있게 설계되어야 함
 - 한 페이지에 너무 많은 색을 사용하여 난잡하지 않도록 설계해야 함
 - 닫기, 이동, 삭제 등과 같은 일반적인 동작에 대한 아이콘은 친숙한 아이콘을 사용해야 함
 - 어떠한 기능을 제공하는 아이콘의 경우 모든 페이지에서 같은(혹은 비슷한) 아이콘을 사용하여 혼란을 줄여야 함
 - 플랫폼과 사용자의 대화 수단인 만큼 플랫폼의 현 상태나 종료 여부를 명확히 표현해야 함
 - 현재 머무르고 있는 페이지에 대한 정보를 표시 해 주어야함
 - 어떠한 기능이 종료되는 시점에는 종료되었다는 표시를 해 주어야함(예, 결제완료 등)
 - 작업 시간이 오래 걸리는 경우 대화상자를 이용해 현재 작업량을 표시해 주어야 함(예, 다운로드 %, 남은 시간 등)

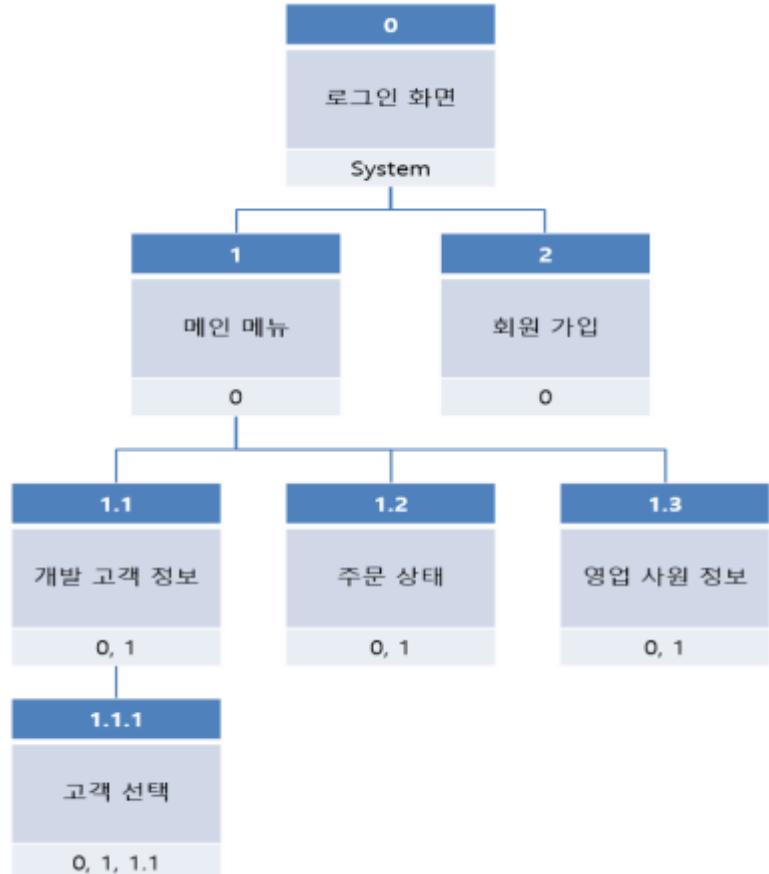
3 단계 : 설계 (물리적 설계 (UI))

✓ UI 설계 프로세스(I)

- 우선 플랫폼을 표현할 수 있는 모든 페이지의 구성물을 표현할 수 있는 대화도를 설계
- 다음으로 설계한 대화도에 구성된 각각의 페이지를 세부적으로 설계하기 위한 **프로토타이핑** 진행

① 스토리 및 대화도(Dialog Diagramming) 작성

- 스토리 및 대화도는 사용자와 플랫폼의 대화를 설계하고 표현한 것
- 대화도는 세 부분으로 나뉜 박스들을 연결하여 작성하며 각 박스들은 상단(화면 참조번호), 중단(화면의 이름 또는 설명), 하단(이전 화면의 참조 번호)으로 구성됨



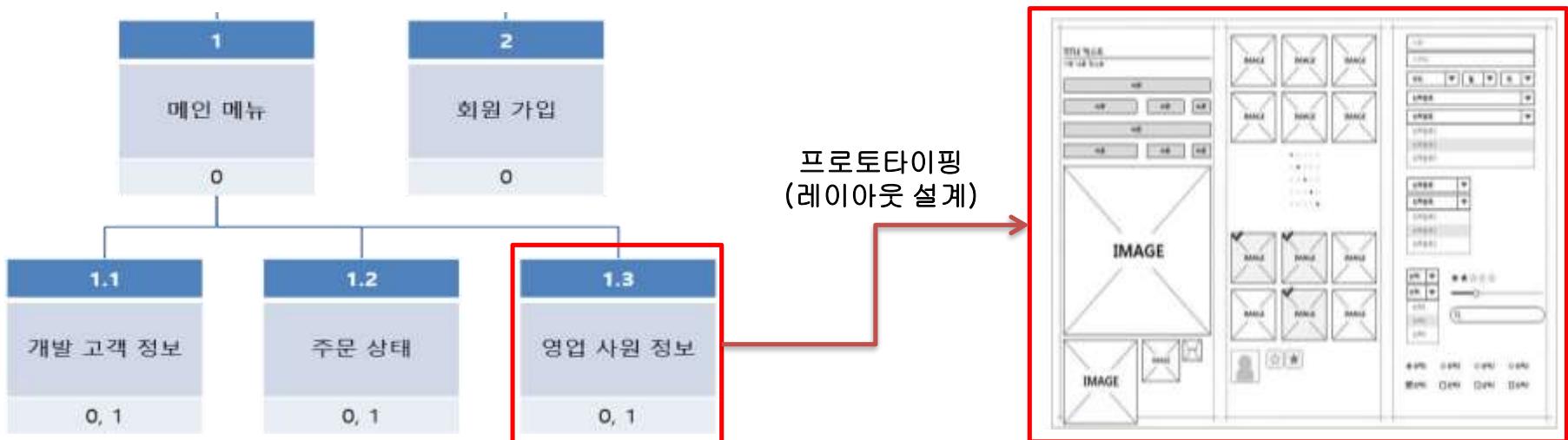
<고객관리 정보시스템 대화도 예시>

3 단계 : 설계 (물리적 설계 (UI))

✓ UI 설계 프로세스(2)

(2) 프로토타이핑(레이아웃 설계)

- 프로토타이핑(레이아웃 설계)은 대화도에서 설계된 각 페이지의 구성을 설계하는 과정으로 다양한 설계 툴을 사용할 수 있음(e.g. Mockups, Wireframe, 카카오 Oven 등 UI 디자인 소프트웨어)
- 플랫폼의 성격에 맞는 디자인을 고려하여 UI 설계 시 고려사항(일관성, 직관성 등)에 따라 작성해야 함



3 단계 : 설계 (물리적 설계 (DB))

- ✓ 데이터 베이스 설계 (1)
 - 논리적 DB 설계 + 물리적 DB 설계
 - 논리적 데이터베이스 설계는 개념적 데이터 모델을 설계하는 단계로 단순한 데이터의 개체와 속성을 표현
 - 물리적 데이터베이스는 논리적 데이터베이스 설계를 바탕으로 각 개체와 속성에 대한 파일 속성이나 저장 방식을 결정
 - 논리적/물리적 데이터 베이스 설계의 수행 방법 및 목적
 - 첫째, 시간이 흘러도 변하지 않고 최소의 중복성을 가지는 안정적인 구조로 데이터를 구조화
 - 둘째, 플랫폼에 존재하는 데이터 요구사항들을 반영하여 논리적인 DB 설계안을 작성
 - 셋째, 논리적 DB를 기반으로 물리적인 DB 설계를 수행
 - 넷째, 데이터베이스 모델을 기술적인 파일 및 데이터베이스 설계안으로 변환
 - 다섯째, 데이터 저장 기술(CD-ROM, 하드디스크, 클라우드 등) 중 효율적으로 정확하고 안전하게 DB를 다룰 수 있는 기술을 선택

3 단계 : 설계 - 물리적 설계 (DB)

✓ 데이터 베이스 설계 (2)

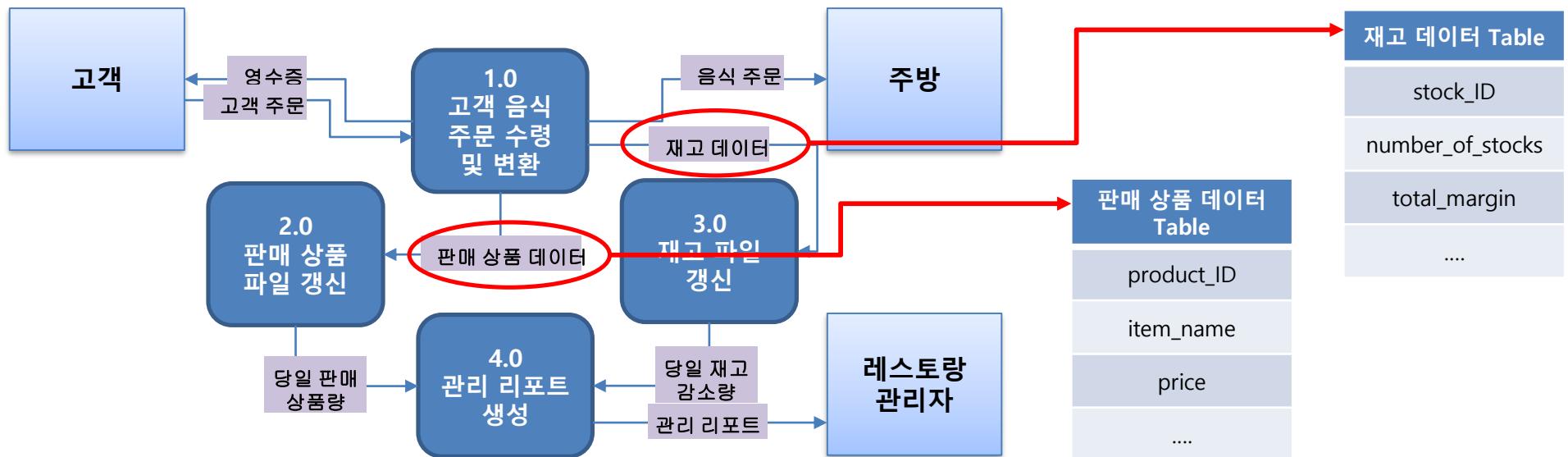
- DB 설계의 최종 목적은 **ERD(Entity-Relationship Diagram)**
- DB 설계의 최종 산출물인 ERD는 DB 설계 결과에서 각 객체의 관계를 도식화
- DB의 설계는 **논리적 데이터베이스 설계** → **물리적 데이터베이스 설계** → **ERD 작성의** 순서로 진행되며 각 설계 단계의 특징은 아래와 같음

논리적 DB 설계	물리적 DB 설계	ERD 작성
<ul style="list-style-type: none">• 데이터 흐름도에 개념적으로 설계된 데이터를 논리적인 데이터 구조로 설계• 논리적 DB는 데이터 테이블, 그리고 테이블을 구성하는 속성들로 구성• 테이블, 속성 이름은 중복되지 않아야 하며 명확해야 함	<ul style="list-style-type: none">• 논리적 DB를 바탕으로 각 속성값들의 Data Type과 Data Size를 정의• 하드웨어나 운영체제의 특성을 고려하여 필요한 인덱스 구조나 내부 저장 구조 등에 대한 물리적 구조를 설계	<ul style="list-style-type: none">• 논리적/물리적 DB 설계를 통해 구조화된 데이터를 구조 및 그에 수반한 제약조건들을 각각의 개체로써 정의하고 관계를 정의하는 Diagram• 각 테이블과 속성의 무결성과 관계를 표현하기 위해 주 키(Primary Key), 외래 키(Foreign Key) 등으로 정의하여 표현하는 기법을 사용

3 단계 : 설계 - 물리적 설계 (DB)

✓ 논리적 데이터베이스 설계

- 논리적 DB 설계에서는 정규화(normalization)라는 절차를 수행
- 정규화는 단순하고 중복성이 없으며 최소의 유지보수가 가능한 성질을 가지는 데이터 모델을 구축하는 과정
 - 데이터 흐름도에서 설계된 개념적 데이터 요구사항들을 하나의 논리적 DB 모델로 통합하는 '뷰통합'을 진행
 - (예: 음식 주문 시스템) 좌측의 데이터 흐름도에 '재고 데이터' 와 '판매 상품 데이터' 등이 개념적 데이터 요구사항으로 정의되어 있으며, 이를 정규화 과정을 통해 하나의 테이블을 생성하고 각 테이블에 맞는 속성들을 정의하여 **논리적 DB를 설계**하였음



3 단계 : 설계 - 물리적 설계 (DB)

✓ 물리적 데이터베이스 설계

- 물리적 데이터베이스는 논리적 데이터베이스 모델에 나타난 각각의 속성들에 대한 **저장 형식(데이터 유형, data type)**을 결정함
 - (예: 음식 주문 시스템) 설계된 논리적 DB에서 정의된 각각의 속성에 맞는 데이터 타입을 정의
 - 데이터 타입은 CHAR, INT, FLOAT, DOUBLE 등 속성값이 가지는 데이터의 특징에 따라 정의할 수 있음

The diagram illustrates the mapping from logical tables to physical tables, specifically focusing on data types. On the left, there are two logical tables: '판매 상품 데이터 Table' and '재고 데이터 Table'. The '판매 상품 데이터 Table' contains columns for 'product_ID', 'item_name', 'price', and an ellipsis. The '재고 데이터 Table' contains columns for 'stock_ID', 'number_of_stocks', and 'total_margin', also with an ellipsis. A large blue arrow points from these logical tables to the right, where they are mapped to physical tables. The '판매 상품 데이터 Table' is mapped to a physical table with columns 'product_ID' (CHAR_10), 'item_name' (CHAR_30), 'price' (CHAR_20), and an ellipsis. The '재고 데이터 Table' is mapped to a physical table with columns 'stock_ID' (CHAR_10), 'number_of_stocks' (INT_10), 'total_margin' (DOUBLE_20), and an ellipsis. Both the original table headers and the mapped physical table headers are highlighted with red boxes.

판매 상품 데이터 Table	재고 데이터 Table	판매 상품 데이터 Table	Data type	재고 데이터 Table	Data type
product_ID	stock_ID	product_ID	CHAR_10	stock_ID	CHAR_10
item_name	number_of_stocks	item_name	CHAR_30	number_of_stocks	INT_10
price	total_margin	price	CHAR_20	total_margin	DOUBLE_20
....

- 효율적인 데이터 접근을 가능케 하는 데이터 저장 매체와 구조를 결정
 - 보다 빠른 데이터 접근을 가능케 하기 위한 주요 구조로는 고유 키(key)와 고유하지 않은 키들에 대한 키인덱스(key index)가 이용됨
 - 오늘날 주요 데이터 저장 매체로는 클라우드 서버나 빅데이터 저장 및 관리가 용이한 대용량 분산처리시스템이 있으며 대표적인 것으로 '하둡(Hadoop)'이 있음

3 단계 : 설계 - 물리적 설계 (DB)

✓ 개체-관계 다이어그램(Entity-Relationship Diagram, ERD)의 작성

- DB 설계의 최종 산출물인 ERD는 DB 설계 결과에서 각 객체의 관계를 도식화한 것
- ERD는 DBMS의 구축에 있어서 반드시 작성되어야 하며, ERD를 작성하는 다양한 툴이 있음 (e.g. ERwin, eXERD, DbDesigner 등)
- ERD 작성 프로세스
 - 개체를 표현 : 개체를 식별할 수 있는 주 키(Primary Key)를 설정(주로 id)하고, 그 외 속성은 주 키를 제외한 모든 속성이 됨
 - 관계성을 표현 : 개체간에 관계성을 나타내기 위해서는 외래 키(Foreign Key)를 이용함
 - 관계 정규화 및 결합 : 1,2 단계에 의해 생성된 관계에서 불필요한 중복성을 제외하기 위한 정규화를 진행. 또한 중복되는 관계를 제거하기 위해 결합과 정규화를 반복

※ ERD 작성 참고

- Erwin : <https://erwin.com/> 에서 Erwin Data Modeler 프로그램을 통해 ERD를 무료로 작성
- eXERD : <http://ko.exerd.com/> 에서 다운로드 가능하며 eclipse 기반의 ERD 작성 툴
- DbDesigner : <http://fabforce.eu/dbdesigner4/> MySQL로 DB를 구축할 시 ERD를 작성하여 바로 MySQL에 적용

3 단계 : 설계 - 물리적 설계 (DB)

✓ (예) 개체-관계 다이어그램(Entity-Relationship Diagram, ERD)

- 각각의 데이터를 구분할 수 있도록 각 테이블에 주 키(PK)를 반드시 설정
- 테이블 간에 관계를 가지기 위해서는 외래 키(FK)를 이용할 수 있음
 - 아래 예시에서는 '관리 리포트 Table'에 '판매 상품 데이터' 및 '재고 데이터'가 종속되어 있는 관계를 표현하였으며 각각의 테이블은 주 키(PK)가 설정되어 있음

관리 리포트 Table		Data type
Manage_report_ID(PK)		CHAR_10
product_id(FK)		CHAR_30
stodk_ID(FK)		CHAR_20
....	

PK : Primary Key, 주 키
FK : Foreign Key, 외래 키

판매 상품 데이터 Table		Data type
product_ID(PK)		CHAR_10
item_name		CHAR_30
price		CHAR_20
....	

재고 데이터 Table		Data type
stock_ID(PK)		CHAR_10
number_of_stocks		INT_10
total_margin		CHAR_20
....	

※ 더 자세한 데이터베이스의 설계 및 ERD 작성 방법은 DB 전문 서적을 참조

❖ ERD design examples

