

### 1. Базы данных. Этапы проектирования.

**База данных** – это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области.

**База данных** – это конкретизированная до экземпляров, объектов и связей модель данных, поддерживаемая на ЭВМ и динамически изменяющаяся в соответствии с изменениями предметной области.

Между собственно данными и пользователями располагается уровень программного обеспечения. Ядром его является система управления базами данных или диспетчер БД.

**Система управления базами данных (СУБД)** - это комплекс программных и языковых средств, необходимых для создания БД, поддержания их в актуальном состоянии и организации поиска в них необходимой информации. Основная функция СУБД – это предоставление пользователю БД возможности работы с ней, не вникая в детали на уровне аппаратного обеспечения. Т.е. все запросы пользователя к БД, добавление и удаление данных, выборки, обновление данных – все это обеспечивает СУБД.

Создавая базу данных, мы стремимся упорядочить информацию по различным признакам для того, чтобы потом извлекать из нее необходимые нам данные в любом сочетании. Сделать это возможно, только если данные структурированы. Структурирование - это набор соглашений о способах представления данных. В зависимости от структуры различают иерархическую, сетевую, реляционную, объектно-ориентированную и гибридную модели баз данных.

**Иерархическая структура базы данных.** Это древовидная структура представления информации. Ее особенность в том, что каждый узел на более низком уровне имеет связь только с одним узлом на более высоком уровне.



Посмотрим, например, на фрагмент иерархической структуры базы данных "Институт": на одной кафедре может работать несколько преподавателей. Такая связь называется "один ко многим" (одна кафедра - много преподавателей). Но если мы попытаемся добавить в эту структуру группы студентов, то нам понадобится связь "многие ко многим": (один преподаватель может работать со многими группами, а одна группа может учиться у многих преподавателей), а такой связи в иерархической структуре быть не может (т.к. связь может быть только с одним узлом на более высоком уровне). Это основной недостаток подобной структуры базы данных.

**Сетевая структура базы данных.** По сути, это расширение иерархической структуры. Все то же самое, но существует связь "многие ко многим". Сетевая структура базы данных позволяет нам добавить группы в наш пример. Недостатком сетевой модели является сложность разработки серьезных приложений.

**Реляционная структура базы данных.** Все данные представлены в виде простых таблиц, разбитых на строки и столбцы, на пересечении которых расположены данные. Эта структура стала настоящим прорывом в развитии баз данных.

**Объектно-ориентированные и гибридные базы данных.** В объектно-ориентированных базах данных данные хранятся в виде объектов, что очень удобно. Но на сегодняшний день такие БД еще распространены, т.к. уступают в производительности реляционным. Гибридные БД совмещают в себе возможности реляционных и объектно-ориентированных, поэтому их часто называют объектно-реляционными. Несомненно, такие БД будут развиваться в будущем, но пока первенство остается за реляционными структурами.

### Этапы проектирования базы данных.

Процесс проектирования включает в себя следующие этапы:

1. **Инфологическое проектирование.** Основными задачами инфологического проектирования являются определение предметной области системы и формирование взгляда на ПО с позиций сообщества будущих пользователей БД. Инфологическая модель ПО представляет собой описание структуры и динамики ПО, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависящих от реализации БД.

2. **Определение требований к операционной обстановке, в которой будет функционировать информационная система.** На этом этапе производится оценка требований к вычислительным ресурсам, необходимым для функционирования системы, определение типа и конфигурации конкретной ЭВМ, выбор типа и версии операционной системы. Объем вычислительных ресурсов зависит от предполагаемого объема проектируемой базы данных и от интенсивности их использования.

3. **Выбор (СУБД) и других инструментальных программных средств.** Выбор СУБД является одним из важнейших моментов в разработке проекта БД, так как он принципиальным образом влияет на весь процесс проектирования БД и реализацию информационной системы.

4. **Логическое проектирование БД.** На этапе логического проектирования разрабатывается логическая структура БД. Решение этой задачи существенно зависит от модели данных, поддерживаемой выбранной СУБД.

5. **Физическое проектирование БД.** Этап физического проектирования заключается в увязке логической структуры БД и физической среды хранения с целью наиболее эффективного размещения данных. Решается вопрос размещения хранимых данных в пространстве памяти, выбора эффективных методов доступа к различным компонентам "физической" БД. Результаты этого этапа документируются в форме схемы хранения на языке определения данных (DDL).

## **2. Оптимизационные задачи в автоматизированных системах технологической подготовки производства.**

### ***Оптимизационные задачи при ТПП:***

1. Найти материал детали, обеспечивающий минимум ее стоимости при выполнении заданных требований.
2. Найти форму и метод изготовления заготовки, обеспечивающие минимум потерь материала.
3. Определить последовательность технологических переходов, обеспечивающую минимальное время изготовления партии деталей.
4. Выбрать оборудование, обеспечивающее:
  - a. минимальную стоимость при удовлетворении требований техпроцесса;
  - b. минимальные приведенные затраты на выполнение технологического контроля;
  - c. минимальный период окупаемости оборудования.

Оптимизационные задачи также могут быть поставлены при:

- программировании станков с ЧПУ;
- выборе метода обработки;
- выборе методов и средств контроля;
- определении требований техники безопасности и обеспечения устойчивости экологической среды и др.

Кроме отдельных оптимизационных задач, рассмотренных выше, в АСТПП, как правило, решается и обобщенная оптимизационная задача: получение ТП, имеющего минимальные затраты на производство единицы продукции. При решении обобщенной задачи учитываются все отдельные критерии путем их суммирования, обобщения, выбора главного критерия и т.д.

### **Оптимизационные задачи, решаемые при автоматизации метода нового планирования.**

Рассмотрим отдельные задачи метода нового планирования.

***Выбор вида заготовки и методов ее изготовления.*** В качестве критериев оптимизации выбора заготовок используют:

- себестоимость изготовления заготовки  $C_z \rightarrow \min$ ;
- себестоимость механической обработки заготовки для получения детали  $C_m \rightarrow \min$ ;
- стоимость отходов металла  $C_o \rightarrow \min$ .

Алгоритм выбора оптимального метода получения заготовки состоит из следующих шагов:

- выбор возможных видов заготовки по материалу детали. В зависимости от вида материала (сталь, чугун, сплавы и т.д.) выбираются методы получения заготовок – отливки, штамповки, прокат, поковки;
- выбор возможных методов изготовления заготовок исходя из серийности детали (единичная, серийная, крупносерийная, массовая); конструктивной формы детали (цилиндрическая, дисковая, пространственная, корпусная и т.д.); массы и размеров детали;
- определение технических характеристик для выбранных видов заготовок (точность, коэффициент использования материала и др.);

- определение себестоимости изготовления заготовки;
- определение себестоимости механической обработки заготовки;
- определение стоимости отходов материала;
- выбор оптимального метода изготовления заготовки для конкретных условий производства.

**Выбор технологических баз.** Алгоритм выбора технологических баз заключается в следующем. После ввода конфигурации детали осуществляется автоматический расчет площадей всех поверхностей детали и их ранжирование в порядке убывания. В качестве основной базы пользователю предлагается поверхность с наибольшей площадью. Если пользователя устраивает данный вариант, то осуществляется переход к выбору вспомогательных баз, если нет – пользователю предлагается следующая по размеру площади поверхность. Выбор вспомогательных баз осуществляется аналогично из поверхностей, оставшихся после выбора основной базы.

**Проектирование технологического маршрута** Данная задача - главная и наиболее трудная. В методе нового планирования используют различные диалоговые подсистемы формирования технологического маршрута. Исходная информация о детали:

- общие сведения;
- сведения о заготовке (поступают из подсистемы выбора заготовки);
- описание наружных и внутренних поверхностей;
- допустимые отклонения.

В диалоговом режиме осуществляется подбор технологических операций, расчет и оптимизация режимов резания, расчет затрат времени на изготовление детали, расчет какого-либо критерия оптимальности (например, себестоимости изготовления детали), оптимизация технологического маршрута по выбранному критерию.

**Проектирование технологических операций.** Каждая технологическая операция, выбранная на этапе проектирования технологического маршрута, проектируется в виде последовательности переходов. Одну и ту же операцию возможно реализовать различной последовательностью отличающихся переходов. Выбор наилучшего варианта осуществляется по критериям: себестоимость операции; время выполнения операции и другим.

**Выбор основного оборудования.** Оборудование для выполнения операций выбирается в зависимости от намеченного состава операций, габаритов и конфигурации детали, требуемой точности обработки, программы выпуска деталей. Состав операции (т.е. перечень поверхностей, обрабатываемых на операции) зависит от возможностей оборудования, и наоборот, оборудование выбирается в зависимости от состава операции, поэтому эти задачи решаются параллельно. Выбор оборудования обычно оптимизируется по критерию стоимости.

**Выбор инструмента.** Выбор режущего инструмента осуществляется для каждого технологического перехода. Исходные данные:

- геометрия детали;
- сведения о заготовке;
- технологические характеристики применяемого оборудования.

Инструмент выбирается из справочной базы, охватывающей все его разновидности.

**Оптимизация проектирования сборочных процессов.** Сборочные работы являются многовариантными как по возможному составу и последовательности

операций техпроцесса, так и по составу применяемой оснастки, оборудования, инструмента. В качестве критериев оптимизации используются:

- трудоемкость процесса сборки;
- технологическая себестоимость;
- цикл сборки (время);
- затраты на сборочную оснастку.

### 3. Таблицы истинности, совершенные нормальные формы представления булевых функций

**Таблица истинности** — это таблица, описывающая логическую функцию.

Под **логической функцией** в данном случае понимается функция, у которой значения переменных и значение самой функции выражают логическую истинность. Например, в двузначной логике они могут принимать значения «истина» либо «ложь» (true либо false, 1 либо 0).

С помощью таблиц истинности можно определять истинностное значение любого высказывания для всех возможных случаев значений истинности составляющих его высказываний. Общее количество всех возможных комбинаций в таблице можно определить по формуле  $N=2^n$ ; где  $N$  - общее число возможных комбинаций,  $n$  - количество входных переменных. В основном таблицы истинности применяются в булевой алгебре и в цифровой электронной технике для описания работы логических схем.

Таблицы истинности в общем случае имеют вид:

1	2	...	n	$f(x_1, x_2, \dots, x_n)$
			.	$f(0, 0, \dots, 0)$
			.	$f(1, 0, \dots, 0)$
			.	$f(0, 1, \dots, 0)$
			.	$f(1, 1, \dots, 0)$
			$\vdots$	$\vdots$
			.	$f(0, 1, \dots, 1)$
			.	$f(1, 1, \dots, 1)$

### Совершенные нормальные формы (СДНФ и СКНФ) записи булевых выражений

**Совершенной дизъюнктивной нормальной формой (СДНФ)** называют форму записи логического выражения, которая представляет собой сумму, каждое слагаемое которой является произведением всех входных аргументов или их инверсий.

СДНФ является избыточной, но логические функции, записанные в СДНФ, легко сравнивать между собой, их удобно преобразовывать в таблицы истинности и составлять по ним карты Карно.

Для того, чтобы получить СДНФ функции, требуется составить её таблицу истинности. К примеру, возьмём таблицу истинности:

$x_1$	$x_2$	$x_3$	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

В ячейках результата  $f(x_1, x_2, x_3)$  отмечаются лишь те комбинации, которые приводят логическое выражение в состояние единицы. Далее рассматриваются значения переменных при которых функция равна 1. Если значение переменной равно 0, то она записывается с инверсией. Если значение переменной равно 1, то без инверсии.

Совершенная ДНФ этой функции:

$$f(x_1, x_2, x_3) = (\overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3}) \vee (\overline{x_1} \wedge \overline{x_2} \wedge x_3) \vee (\overline{x_1} \wedge x_2 \wedge \overline{x_3}) \vee (x_1 \wedge x_2 \wedge \overline{x_3})$$

В некоторых случаях более удобной формой записи логического выражения является **совершенная конъюнктивная нормальная форма (СКНФ)**. Это произведение сомножителей, каждый из которых является суммой всех входных аргументов или их инверсий. Так же, как и СДНФ, СКНФ является явно избыточной.