

STRUTTURA GENERALE DELL'ESAME

L'esame di Stato di Informatica ITIS è composto da:

- **PRIMA PARTE:** Analisi e progettazione sistema informativo (obbligatoria)
- **SECONDA PARTE:** Quesiti a scelta (scegli 2 su 4)

Durata: 6 ore **Linguaggi:** SQL, HTML, CSS, PHP, JavaScript

PROCEDIMENTO PRIMA PARTE (4 ORE)

1. ANALISI DEL PROBLEMA (20 minuti)

```
SCENARIO: [2-3 righe del contesto]
ATTORI: [Chi usa il sistema]
FUNZIONI: [Cosa deve fare il sistema]
DATI: [Quali informazioni gestire]
```

2. SCHEMA E-R (30 minuti)

Esempio Scuola Educational Games:

```
ENTITÀ:
- DOCENTE (id, nome, cognome, email)
- CLASSE (codice, nome, materia, link)
- STUDENTE (id, nome, cognome, email)
- VIDEOGIOCO (id, titolo, argomento, descrizione)
- MONETA (id, valore, descrizione)

RELAZIONI:
- DOCENTE crea CLASSE (1:N)
- STUDENTE si iscrive CLASSE (N:M)
- CLASSE usa VIDEOGIOCO (N:M)
- STUDENTE raccoglie MONETA (N:M)
```

3. SCHEMA LOGICO (30 minuti)

```
CREATE DATABASE esame2025;
USE esame2025;
```

```

CREATE TABLE docenti (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(50),
    cognome VARCHAR(50),
    email VARCHAR(100)
);

CREATE TABLE classi (
    codice VARCHAR(10) PRIMARY KEY,
    nome VARCHAR(50),
    materia VARCHAR(50),
    link VARCHAR(200),
    id_docente INT,
    FOREIGN KEY (id_docente) REFERENCES docenti(id)
);

CREATE TABLE studenti (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(50),
    cognome VARCHAR(50),
    email VARCHAR(100)
);

CREATE TABLE iscrizioni (
    id_studente INT,
    codice_classe VARCHAR(10),
    data_iscrizione DATE,
    PRIMARY KEY (id_studente, codice_classe),
    FOREIGN KEY (id_studente) REFERENCES studenti(id),
    FOREIGN KEY (codice_classe) REFERENCES classi(codice)
);

```

4. CODICE PHP SEMPLIFICATO (2.5 ore)

FILE: config.php

```

<?php
session_start();
$conn = new mysqli("localhost", "root", "", "esame2025");
if ($conn->connect_error) {
    die("Errore: " . $conn->connect_error);
}
?>

```

FILE: index.php

```

<!DOCTYPE html>
<html>
<head>
  <title>Sistema Gestionale</title>
  <style>
    body { font-family: Arial; margin: 20px; }
    input, select { width: 200px; padding: 5px; margin: 5px; }
    button { padding: 10px; background: blue; color: white; border:
none; }
    table { border-collapse: collapse; width: 100%; }
    th, td { border: 1px solid black; padding: 8px; }
  </style>
</head>
<body>
  <h1>Sistema [Nome Progetto]</h1>

  <a href="inserisci.php">Inserisci Dati</a> |
  <a href="visualizza.php">Visualizza</a> |
  <a href="ricerca.php">Ricerca</a>

  <h2>Benvenuto nel sistema</h2>
  <p>Seleziona un'operazione dal menu sopra.</p>
</body>
</html>

```

FILE: inserisci.php

```

<?php include 'config.php'; ?>
<!DOCTYPE html>
<html>
<head>
  <title>Inserimento Dati</title>
  <style>
    body { font-family: Arial; margin: 20px; }
    input { width: 200px; padding: 5px; margin: 5px; }
    button { padding: 10px; background: green; color: white; border:
none; }
  </style>
</head>
<body>
  <h1>Inserisci Nuovo Record</h1>

  <?php
if (isset($_POST['inserisci'])) {
  $nome = $_POST['nome'];
  $cognome = $_POST['cognome'];
  $email = $_POST['email'];

```

```

        if (!empty($nome) && !empty($cognome)) {
            $sql = "INSERT INTO docenti (nome, cognome, email)
                VALUES ('$nome', '$cognome', '$email')";

            if ($conn->query($sql)) {
                echo "<p style='color:green'>Inserimento completato!</p>";
            } else {
                echo "<p style='color:red'>Errore: " . $conn->error . "
</p>";
            }
        } else {
            echo "<p style='color:red'>Compila tutti i campi
obbligatori</p>";
        }
    }
    ?>

<form method="POST">
    <p>Nome: <input type="text" name="nome" required></p>
    <p>Cognome: <input type="text" name="cognome" required></p>
    <p>Email: <input type="email" name="email"></p>
    <p><button type="submit" name="inzerisci">Inzerisci</button></p>
</form>

<a href="index.php">Torna al menu</a>
</body>
</html>

```

FILE: visualizza.php

```

<?php include 'config.php'; ?>
<!DOCTYPE html>
<html>
<head>
    <title>Visualizza Dati</title>
    <style>
        body { font-family: Arial; margin: 20px; }
        table { border-collapse: collapse; width: 100%; }
        th, td { border: 1px solid black; padding: 8px; }
        th { background-color: lightgray; }
    </style>
</head>
<body>
    <h1>Elenco Docenti</h1>

    <?php
    $sql = "SELECT * FROM docenti ORDER BY cognome";
    $result = $conn->query($sql);

```

```

        if ($result->num_rows > 0) {
            echo "<table>";
            echo "<tr><th>ID</th><th>Nome</th><th>Cognome</th><th>Email</th>
<th>Azioni</th></tr>";

            while ($row = $result->fetch_assoc()) {
                echo "<tr>";
                echo "<td>" . $row['id'] . "</td>";
                echo "<td>" . $row['nome'] . "</td>";
                echo "<td>" . $row['cognome'] . "</td>";
                echo "<td>" . $row['email'] . "</td>";
                echo "<td>
                    <a href='modifica.php?id=" . $row['id'] .
''>Modifica</a> |
                    <a href='cancella.php?id=" . $row['id'] .
''>Cancella</a>
                </td>";
                echo "</tr>";
            }
            echo "</table>";
        } else {
            echo "<p>Nessun dato presente.</p>";
        }
    ?>

    <p><a href="index.php">Torna al menu</a></p>
</body>
</html>

```

FILE: ricerca.php

```

<?php include 'config.php'; ?>
<!DOCTYPE html>
<html>
<head>
    <title>Ricerca</title>
    <style>
        body { font-family: Arial; margin: 20px; }
        input { width: 200px; padding: 5px; margin: 5px; }
        table { border-collapse: collapse; width: 100%; }
        th, td { border: 1px solid black; padding: 8px; }
    </style>
</head>
<body>
    <h1>Ricerca Docenti</h1>

    <form method="POST">

```

```

        <p>Cerca: <input type="text" name="cerca" placeholder="Nome o
cognome">
        <button type="submit">Cerca</button></p>
    </form>

    <?php
    if (isset($_POST['cerca'])) {
        $termine = $_POST['cerca'];

        $sql = "SELECT * FROM docenti
                WHERE nome LIKE '%$termine%' OR cognome LIKE '%$termine%'";
        $result = $conn->query($sql);

        if ($result->num_rows > 0) {
            echo "<h3>Risultati trovati:</h3>";
            echo "<table>";
            echo "<tr><th>Nome</th><th>Cognome</th><th>Email</th></tr>";

            while ($row = $result->fetch_assoc()) {
                echo "<tr>";
                echo "<td>" . $row['nome'] . "</td>";
                echo "<td>" . $row['cognome'] . "</td>";
                echo "<td>" . $row['email'] . "</td>";
                echo "</tr>";
            }
            echo "</table>";
        } else {
            echo "<p>Nessun risultato trovato.</p>";
        }
    }
    ?>

    <p><a href="index.php">Torna al menu</a></p>
</body>
</html>

```

FUNZIONI ESSENZIALI SEMPRE RICHIESTE

1. GENERAZIONE CODICE CASUALE

```

function generaCodice($lunghezza = 6) {
    $caratteri = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
    $codice = '';
    for ($i = 0; $i < $lunghezza; $i++) {
        $codice .= $caratteri[rand(0, strlen($caratteri) - 1)];
    }
}

```

```
        return $codice;
    }

    // Uso:
    $codice_classe = generaCodice();
    echo $codice_classe; // Es: AB4K7M
```

2. CALCOLO STATISTICHE

```
function calcolaMedia($tabella, $campo) {
    global $conn;
    $sql = "SELECT AVG($campo) as media FROM $tabella";
    $result = $conn->query($sql);
    $row = $result->fetch_assoc();
    return round($row['media'], 2);
}

// Uso:
$media_voti = calcolaMedia('studenti', 'voto');
```

3. JOIN TRA TABELLE

```
// Esempio: studenti iscritti per classe
$sql = "SELECT c.nome as classe, s.nome, s.cognome
        FROM classi c
        JOIN iscrizioni i ON c.codice = i.codice_classe
        JOIN studenti s ON i.id_studente = s.id
        ORDER BY c.nome, s.cognome";
```

GESTIONE TEMPO (6 ORE)

Tempo	Attività
0-20 min	Lettura e analisi traccia
20-50 min	Schema E-R e progettazione
50-80 min	Schema logico e creazione DB
80-230 min	Codifica PHP (2h 30min)
230-290 min	Seconda parte (2 quesiti)
290-360 min	Controllo e test finale

SECONDA PARTE - QUESITI TIPICI

Query SQL Complesse

```
-- Conta studenti per classe
SELECT c.nome, COUNT(i.id_studente) as totale_studenti
FROM classi c
LEFT JOIN iscrizioni i ON c.codice = i.codice_classe
GROUP BY c.codice, c.nome;

-- Docenti con più di 2 classi
SELECT d.nome, d.cognome, COUNT(c.codice) as num_classi
FROM docenti d
JOIN classi c ON d.id = c.id_docente
GROUP BY d.id
HAVING COUNT(c.codice) > 2;
```

Algoritmi Base

```
// Ordinamento array
function ordinaArray($array) {
    sort($array);
    return $array;
}

// Ricerca in array
function cercaValore($array, $valore) {
    return array_search($valore, $array);
}
```

ERRORI DA EVITARE

Database:

- Non dimenticare `AUTO_INCREMENT` per le chiavi primarie
- Sempre usare `FOREIGN KEY` per le relazioni
- Nomi tabelle e campi senza spazi

PHP:

- Sempre includere `config.php` all'inizio
- Controllare se `$_POST` è impostato prima di usarlo
- Chiudere i tag HTML correttamente

Sicurezza Base:

- Controllare campi vuoti prima di inserire
 - Usare `htmlspecialchars()` quando stampi dati utente
-

TEMPLATE VELOCE TRACCIA TIPO

Se la traccia parla di: Scuola/Studenti/Corsi **Entità principali:** STUDENTE, CORSO, DOCENTE, ISCRIZIONE **Relazioni:** Studente si iscrive Corso (N:M), Docente insegna Corso (1:N)

Se la traccia parla di: E-commerce/Prodotti/Ordini

Entità principali: CLIENTE, PRODOTTO, ORDINE, DETTAGLIO_ORDINE **Relazioni:** Cliente effettua Ordine (1:N), Ordine contiene Prodotto (N:M)

Se la traccia parla di: Biblioteca/Libri/Prestiti **Entità principali:** UTENTE, LIBRO, PRESTITO **Relazioni:** Utente prende Prestito (1:N), Prestito riguarda Libro (N:1)

CHECKLIST FINALE

Prima di consegnare:

- ☐ Database creato e funzionante
- ☐ Almeno 3 pagine PHP (inserisci, visualizza, ricerca)
- ☐ Menu di navigazione tra le pagine
- ☐ Form di inserimento con controllo errori
- ☐ Almeno una query con JOIN
- ☐ Stile CSS di base applicato
- ☐ Codice commentato dove necessario
- ☐ 2 quesiti della seconda parte risolti

IMPORTANTE: Testa sempre il codice! Verifica che ogni pagina si carichi senza errori e che le operazioni funzionino correttamente.